



**Rancang Bangun Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktifitas
Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG
Shield**

PROYEK AKHIR

**Diajukan kepada Fakultas Teknik Universitas Negeri Yogyakarta untuk
Memenuhi Sebagian Persyaratan Guna Memperoleh
Gelar Ahli Madya Teknik**



**ERI AGSIS SATRIO WIBISONO
NIM. 14507134032**

**PROGRAM STUDI TEKNIK ELEKTRONIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA
2018**

LEMBAR PERSETUJUAN

PROYEK AKHIR

**Rancang Bangun Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktifitas
Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG
Shield**

Oleh

ERI AGSIS SATRIO WIBISONO


14507134032


Telah memenuhi syarat dan disetujui oleh Dosen Pembimbing untuk
dilaksanakan Ujian Akhir Proyek Akhir bagi yang
bersangkutan

Yogyakarta, 23 Juli 2018

Mengetahui,
Kaprodi Teknik Elektronika

Menyetujui,
Pembimbing Proyek Akhir


Dr. Sri Waluyanti, M.Pd.
NIP. 19581218 198603 2 00


Dr. Fatchul Arifin, M.T.
NIP. 19720508 198802 1 002

LEMBAR PENGESAHAN

PROYEK AKHIR

**Rancang Bangun Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas
Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG
Shield**

Dipersiapkan dan Disusun Oleh:

ERI AGSIS SATRIO WIBISONO

14507135032

Telah dipertahankan di depan Tim Penguji Proyek Akhir Progam Studi

Teknik Elektronika Fakultas Teknik Universitas Negeri

Yogyakarta

Pada tanggal Juli 2018

SUSUNAN DEWAN PENGUJI

Nama	Jabatan	Tanda Tangan	Tanggal
1. Dr. Fatchul Arifin, M.T.	Ketua Penguji		16/08-2018
2. Ahmad Awaluddin Baiti, S.Pd.T., M.Pd.	Sekretaris Penguji		16/8 18
3. Dr.Masduki Zakarijah, M.T.	Penguji		10/08/2018

Yogyakarta, Juli 2018

Dekan Fakultas Teknik UNY


Dr. Widarto, M.Pd.

NIP. 19631230 198812 1 001

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini :

Nama : Eri Agsis Satrio Wibisono

NIM : 14507135032

Progam Studi : Teknik Elektronika D3

Judul Proyek akhir : Rancang Bangun Alat Elektrokardiograf (EKG)
Sebagai Pemantau Aktivitas Kelistrikan Jantung
Dengan Menggunakan Arduino Mega 2560 dan EKG
Shield

Menyatakan bahwa Proyek Akhir ini benar-benar hasil pekerjaan saya sendiri.
Sepanjang pengetahuan saya, tidak terdapat karya atau pendapat yang ditulis atau
diterbitkan orang lain kecuali sebagai acuan kutipan dengan mengikuti tata
penulisan karya ilmiah yang lazim.

Yogyakarta, 23 Juli 2018

Yang menyatakan,



Eri Agsis Satrio Wibisono
NIM . 14507134032

MOTTO

“Man Jadda Wajada”

Siapa Yang Bersungguh-sungguh Pasti Akan Berhasil

Jalani apa adanya, karena semua akan indah pada waktunya

Be Humble

“Man Shobaro Zhofiro”

Barang Siapa Yang Sabar Akan Beruntung

LEMBAR PERSEMBAHAN

Rasa Syukur dan Sujud kepada Allah SWT atas segala rahmat dan kesempatan untuk menyelesaikan tugas akhir dengan segala kekuranganku. Segala syukur telah aku ucapkan karena telah memberiku segala kekuatan untuk menyelesaikan tugas akhir ini, menamparku dikala aku mulai bimbang dan putus asa. Atas karunia serta kemudahan yang Engkau berikan, akhirnya Laporan Proyek Akhir ini dapat terselesaikan. Sholawat dan salam selalu terlimpahkan kepada junjungan kita Nabi Besar Muhammad SAW.

Kupersembahkan karyaku ini kepada orang yang sangat kukasihi dan kusayangi, ibunda dan ayah. Rasa terimakasih mungkin belum seberapa dibanding dengan mereka yang terus semangat membiayai serta membekali ilmu. Namun, semoga perjuangan ini bisa dapat setidaknya membuat mereka tersenyum bahagia. Teruntuk adikku, tetap semangat dalam menjalani hari dan selalu semangat serta tekun dalam menuntut ilmu di jenjang kuliahmu. Untuk kakakku semoga menjadi ibu yang baik dan bisa menjadi contoh bagi anakmu kelak. Semoga tulisan ini bisa mencapai dan menggapai apa yang diinginkan ayah dan ibu selama ini dan menjadi sebuah batu loncatan untuk membuat adikku menjadi semangat menggapai cita-citanya. Terimakasih karena selalu membuatku termotivasi. Semoga karya ini bisa berguna sebagaimana mestinya.

Untuk mu teman-temanku di kelas B Teknik Elektronika 2014, kebersamaanmu sungguh takkan pernah sirna walau terkikis oleh waktu. Kebersamaan kalian telah membuatku sadar bahwa pertemanan bukan hanya sebatas tertawa dan bercanda, tapi lebih dari itu. sungguh, kebersamaan yang kita

bangun selama ini telah banyak merubah kehidupanku. Mungkin beberapa tahun lagi banyak dari kita yang akan lupa tentang kenangan ini, *but not for me*. Bila Tuhan memberikan ku umur panjang, akan aku bagi harta yang tak ternilai ini (persahabatan) dengan anak dan cucuku kelak.

Seluruh Dosen Pengajar Teknik Elektronika, terima kasih banyak untuk semua ilmu, didikan dan pengalaman yang sangat berarti yang telah kalian berikan. Untuk Dosen Pembimbing saya Dr.Fatchul Arifin, S.T., M.T. terima kasih bimbingan dan bantuannya selama ini, atas nasihat dan pelajaran yang saya dapatkan, saya tidak akan lupa atas bantuan dan kesabaran dari Bapak.

“Ya Allah, Berikanlah Ilmu Yang Bermanfaat Bagi HambaMu ini, Untuk Menunjang sebuah Kehidupan Di Depan Kelak, Serta Untuk Sebuah Langkah Keberhasilan”

PROYEK AKHIR

Rancang Bangun Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield

Oleh : Eri Agsis Satrio Wibisono

NIM : 14507134032

ABSTRAK

EKG shield merupakan komponen utama dalam mengubah sinyal listrik yang dihasilkan oleh jantung, menjadi sinyal digital yang nantinya diolah oleh arduino mega dan ditampilkan hasilnya pada aplikasi di komputer. Tujuan proyek akhir ini adalah untuk merancang dan membangun alat Elektrokardiograf (EKG) sebagai pemantau aktivitas kelistrikan jantung dengan menggunakan arduino mega 2560 dan EKG shield.

Perancangan alat EKG sebagai pemantau aktifitas kelistrikan jantung dengan menggunakan arduino mega 2560 dan EKG shield terdiri dari beberapa tahapan yaitu identifikasi kebutuhan, analisis kebutuhan, blok diagram rangkaian, perancangan sistem, langkah pembuatan alat, diagram alir program, pengujian alat, dan pengambilan data. Alat ini menggunakan mikrokontroller arduino mega 2560, elektroda sebagai sensor yang berfungsi untuk mengambil sinyal listrik dari jantung, EKG shield OLIMEX sebagai *sub-processing* yang berfungsi memproses sinyal listrik dari elektroda, dan software FreeHC digunakan untuk menampilkan hasil gelombang jantung yang telah diproses.

Hasil pengujian perancangan alat EKG sebagai pemantau aktifitas kelistrikan jantung ini mampu menampilkan gelombang jantung dengan cukup baik. Setelah dibandingkan dengan alat kalibrasi sinyal jantung di rumah sakit, didapat terdapat kesamaan lebih dari 60% pada gelombang R dan T, sedangkan untuk gelombang P, Q, dan S kesamaan gelombang kurang dari 35%, dan pada pengujian pengukuran BPM terdapat error 5%. Secara keseluruhan alat sudah bisa berfungsi dengan baik dan menampilkan gelombang jantung sebagaimana mestinya

Kata kunci : *Elektrokardiograf, Jantung, EKG shield OLIMEX, Arduino, Gelombang PQRS*

KATA PENGANTAR

Assalamu'alaikum wr. wb.

Puji syukur penulis panjatkan kehadirat Allah SWT sehingga dengan rahmat dan hidayah-Nya Laporan Proyek Akhir ini dapat terselesaikan tanpa halangan yang berarti. Sholawat serta salam tercurah pada Qudwah kita Rasulullah SAW keluarga, sahabat dan orang-orang yang istiqomah di jalan-Nya.

Dalam menyusun Laporan Proyek Akhir ini penulis merasa banyak kekurangan karena terbatasnya kemampuan dan pengetahuan penulis. Oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Dr.Fatchul Arifin, S.T., M.T. selaku Ketua Jurusan Pendidikan Teknik Elektronika Universitas Negeri Yogyakarta sekaligus Dosen Pembimbing Penyusun Laporan Proyek akhir.
2. Dra. Sri Waluyanti, M.Pd. selaku Ketua Program Studi Diploma III dan Koordinator Proyek Akhir Jurusan Pendidikan Teknik Elektronika Universitas Negeri Yogyakarta.
3. Dr. Widarto, M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
4. Seluruh Dosen Pengajar dan karyawan Teknik Elektronika Universitas Negeri Yogyakarta atas bekal ilmu yang diberikan kepada penulis.
5. Orangtua yang telah mendidik hingga tumbuh dewasa, memberi pendidikan serta usahanya tak kenal lelah dalam membiayai apapun dan penuh dengan kesabaran yang tak bisa diukur

6. Kepada Mas Purwanto selaku kakak tingkat, pembimbing, pemberi pertimbangan, dan masukan dalam proyek akhir ini, dari awal hingga selesainya proyek ini.
7. Teman-teman Fakultas Teknik UNY khususnya Teknik Elektronika kelas B 2014 yang telah memberikan bantuan sehingga pembuatan proyek akhir ini dapat terselesaikan. Terutuk Ilham Hamiddin dan Muhammad Nadhil yang selalu menemani dalam proses penyelesaian proyek ini.
8. Pak Agus Susilo selaku ketua Ikatemi yang sudah memberikan masukan dan membagi wawasannya terkait proyek ini
9. Semua pihak yang tidak dapat saya sebutkan satu persatu yang telah membantu hingga terselesainya laporan dan alat ini.

Penulis menyadari bahwa laporan ini masih banyak kekurangan dan jauh dari sempurna walaupun penulis telah berusaha untuk mendekati kesempurnaan, maka penulis berharap para pembaca memberikan saran dan kritik yang membangun.

Akhir kata penulis mohon maaf yang sebesar-besarnya apabila ada kekeliruan di dalam penulisan laporan ini.

Wassalamu'alaikum wr. wb.

Yogyakarta, 23 Juli 2018

Penulis,

Eri Agsis Satrio Wibisono

DAFTAR ISI

PROYEK AKHIR	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR PERNYATAAN KEASLIAN	iv
MOTTO	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
DAFTAR LAMPIRAN	xvii
BAB I	1
PENDAHULUAN	1
A. Latar Belakang Masalah	1
B. Identifikasi Masalah	3
C. Batasan Masalah	3
D. Rumusan Masalah	3
E. Tujuan	4
F. Manfaat	5
G. Keaslian Gagasan	6
BAB II	8
PENDEKATAN PEMECAHAN MASALAH	8
A. Jantung	8
B. Elektrofisiologis Sel Jantung	10
C. Konduksi Jantung	13
D. EKG (Elektrokardiograf)	15
E. Titik Sadapan Sinyal (lead) EKG	18
F. Pembacaan EKG	19

G. Elektroda	21
H. EKG Shield Olimex	22
I. Arduino	23
J. FreeHC	29
BAB III	31
KONSEP PERANCANGAN	31
A. Identifikasi Kebutuhan	31
B. Analisis Kebutuhan	32
C. Blok Diagram sistem	34
D. Perancangan Sistem	35
E. Langkah Pembuatan Alat	37
1. Pembuatan Box	37
2. Pemasangan Rangkaian Pada Box	39
F. <i>Software</i>	39
1. Algoritma Program	39
2. Flow chart	39
G. Spesifikasi Alat	41
H. Pengujian Alat	41
I. Tabel Hasil Uji	42
J. Pengoperasian Alat	44
BAB IV	45
PENGUJIAN DAN PEMBAHASAN	45
A. Uji Fungsional	45
B. Uji Unjuk Kerja	49
C. Pembahasan	52
D. Cara Kerja Alat	58
BAB V	59
KESIMPULAN DAN SARAN	59
A. Kesimpulan	59
B. Keterbatasan Alat	60
C. Penelitian Lanjutan	60

DAFTAR PUSTAKA	61
LAMPIRAN.....	62

DAFTAR TABEL

Tabel 1. Tipe Data Bahasa C.....	27
Tabel 2. Operasi Kondisi.....	28
Tabel 3. Pengujian Tampilan Modul EKG di PC	42
Tabel 4. Perbandingan Pengujian EKG	42
Tabel 5. Pengujian Keakuratan Pembacaan BPM Alat.....	43
Tabel 6. Pengujian Unjuk Kerja Sistem.....	43
Tabel 7. Pengujian Tampilan Modul EKG di PC	45
Tabel 8. Perbandingan Pengujian EKG	46
Tabel 9. Persamaan Hasil Gelombang PQRST	49
Tabel 10. Pengujian Keakuratan Pembacaan BPM Alat.....	40
Tabel 11. Uji Alat Keseluruhan	51

DAFTAR GAMBAR

Gambar 1. Jantung Manusia	8
Gambar 2. Potensial Aksi Jantung	12
Gambar 3. Struktur Konduksi Jantung.....	13
Gambar 4. Potensial Membran Pemicu Jantung	15
Gambar 5. Sinyal Keluaran EKG.....	16
Gambar 6. Titik Referensi Pemasangan Elektroda 3 Lead	19
Gambar 7. Sinyal EKG	20
Gambar 8. <i>Electrode Snap Cable</i>	21
Gambar 9. <i>EKG Gel Electrodes</i>	22
Gambar 10. EKG Shield Olimex	23
Gambar 11. Arduino Mega 2560	25
Gambar 12. Tampilan Arduino IDE.....	26
Gambar 13. Program FreeHC	30
Gambar 14. Blok Diagram Sistem	34
Gambar 15. Desain Box	38
Gambar 16. Flowchart Sistem	40

DAFTAR LAMPIRAN

Lampiran 1. Gambar Alat	64
Lampiran 2. Foto Pemasangan Elektroda pada Responden	65
Lampiran 3. Desain Box	66
Lampiran 4. Source Code	67
Lampiran 5. Scan Form Kuesioner	72
Lampiran 6. Datasheet Arduino Mega 2560	73
Lampiran 7. Datasheet EKG Shield Olimex	81

BAB I

PENDAHULUAN

A. Latar Belakang

Medical Electronics Engineering atau dapat juga disebut teknik elektromedik, merupakan sebuah bidang studi yang mempelajari ilmu elektronika dan medis. Dokter dapat memeriksa berbagai keadaan pasien dan menyembuhkan hampir semua penyakit dibantu dengan menggunakan peralatan medis yang modern dan teliti, sedangkan teknik elektromedik dapat dikatakan sebagai dokter bagi alat-alat yang dipakai untuk memeriksa pasien, yang mana setiap dokter tidak bisa berbuat banyak jika alat tersebut rusak. Sehingga dapat dikatakan, teknik elektromedik mempelajari mekanisme, pengoperasian, dan perawatan terhadap alat-alat medis. (myclassroom, 2012)

Di Universitas Negeri Yogyakarta (UNY) sendiri, Pada Prodi Pendidikan Teknik Elektronika dan Informatika (JPTEI), Teknik Elektromedis mulai diberlakukan pada tahun ajaran 2016/2017, yang berdasar pada Surat Keputusan DirJen Dikdasmen Kemdikbub no. 4678/D/KEP/MK/2016 yang menjelaskan tentang kompetensi keahlian Instrumentasi Medik yang termasuk dalam program keahlian Teknik Elektronika. (jppei, 2016)

Mata kuliah IEM di JPTEI yang diselenggarakan, masih memerlukan pengembangan. Hal ini selaras dengan keadaan yang disampaikan oleh dosen pengampu mata kuliah tersebut, Bu Dessy Irawati, M.T., beliau menyampaikan bahwa modul, *jobsheet*, dan instrumen praktik masih perlu untuk dikembangkan.

Referensi materi kuliah yang diajarkan masih mengambil dari *E-book Biomedical instrumentations* dan materi-materi praktik yang tersedia di internet.

Berdasarkan paragraf diatas, salah satu kondisi yang dipandang perlu ada pembuatan dan pengembangan adalah instrumen praktikum. Pengertian instrumen praktikum sendiri ialah alat atau modul yang dipakai untuk melancarkan proses pembelajaran suatu mata kuliah, terkhusus mata kuliah elektromedis. Sehingga dalam hal ini dibutuhkan sebuah instrumen praktikum untuk mendukung mata kuliah elektromedis agar lebih memudahkan mahasiswa untuk memahami bidang yang sedang dipelajarinya.

Modul Elektrokardiograf (EKG) merupakan salah satu modul elektromedis yang dapat dikembangkan sesuai kebutuhan. Arsitektur hardware dari modul ini menyesuaikan secara default dengan arduino, yang mana arduino ini, sering dipakai oleh para developer software maupun hardware. (Arduino, 2018). Selain itu modul ini cocok digunakan bagi para pemula untuk mengenal lebih jauh mata kuliah elektromedis, dimana dalam modul ini, mahasiswa dapat mengerti pemanfaatan teknologi elektromedis di jaman sekarang ini.

Dari beberapa permasalahan diatas, diperlukan suatu modul berupa instrumen praktik tentang EKG, yang dapat digunakan sebagai salah satu modul praktik yang diharapkan dapat membuat mahasiswa lebih memahami mata kuliah elektromedis. Modul EKG yang digunakan sudah terbilang modern. Menurut KBBI (Kamus Besar Bahasa Indonesia), modern adalah suatu hal yang baru atau mutakhir. Sehingga dapat disimpulkan modul EKG ini merupakan sebuah hal yang mutakhir dalam dunia elektronika, dapat dilihat dari pemakaian komponen dalam modul yang

berjenis SMD (*Surface Mount Technology*), dan penggunaan jalur PCB dalam modul yang sudah menggunakan jenis 2 layer.

B. Identifikasi Masalah

Berdasarkan uraian latar belakang masalah di atas, maka dapat dibuat identifikasi masalah sebagai berikut :

1. Minimnya tenaga elektromedis di Indonesia
2. Mata kuliah IEM yang diselenggarakan masih mempunyai kekurangan dalam hal modul, jobsheet, dan instrumen praktik
3. Tidak adanya instrumen praktik yang berkaitan dengan EKG

C. Batasan Masalah

Berdasarkan identifikasi masalah diatas permasalahan dibatasi pada poin 3, yaitu tidak adanya instrumen praktik yang berkaitan dengan EKG. Rancang Bangun instrumen praktik EKG ini, akan dibuat menggunakan Arduino MEGA 2560 yang akan menjadi otak utama dari alat, dan modul EKG shield OLIMEX yang akan memproses sinyal listrik dari jantung untuk diolah lebih lanjut lewat Arduino MEGA 2560. Pembatasan ini didasarkan atas keterbatasan dan bertujuan untuk meminimalisir pengetahuan bidang ilmu yang tidak dikuasai.

D. Rumusan Masalah

Berdasarkan berbagai hal yang telah dikemukakan diatas, maka rumusan masalah yang akan dikaji dalam pembuatan proyek akhir ini adalah :

1. Bagaimana perancangan Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield ?
2. Bagaimana merealisasikan *software* Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield ?
3. Bagaimana unjuk kerja Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield ?

E. Tujuan Proyek Akhir

Tujuan dari pembuatan proyek akhir yang berjudul “Rancang Bangun Alat Elektrokardiograf (EKG) Untuk Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield” untuk memenuhi persyaratan guna memperoleh gelar ahli madya adalah sebagai berikut :

1. Merealisasikan perancangan *Hardware* Elektrokardiograf (EKG) Untuk Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield.
2. Merealisasikan rancangan *software* Alat Elektrokardiograf (EKG) Untuk Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield.

3. Mengetahui unjuk kerja Alat Elektrokardiograf (EKG) Untuk Pemantau Aktifitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield.

F. Manfaat Proyek Akhir

Dalam pembuatan proyek akhir ini, diharapkan dapat memberikan manfaat sebagai berikut :

1. Bagi Mahasiswa
 - a. Sebagai sarana implementasi pengetahuan yang didapatkan saat di bangku pendidikan.
 - b. Mampu merealisasikan teori yang didapatkan selama mengikuti perkuliahan.
 - c. Sebagai wujud kontribusi terhadap Universitas baik dalam citra maupun daya tawar terhadap masyarakat luas.
 - d. Memberikan motivasi kepada mahasiswa untuk tetap berkarya dan menjadi salah satu pelaku dalam kemajuan teknologi di zaman modern ini.
2. Bagi Jurusan Pendidikan Teknik Elektronika
 - a. Terciptanya alat yang inovatif dan bermanfaat sebagai sarana ilmu pengetahuan.
 - b. Sebagai wujud partisipasi dalam pengembangan di bidang ilmu dan teknologi.

- c. Sebagai tolak ukur daya serap mahasiswa yang bersangkutan selama menempuh pendidikan dan kemampuan ilmunya secara praktis.
3. Bagi Masyarakat, Peneliti, dan Dunia Industri
- a. Terciptanya alat sebagai sarana peningkatan teknologi dalam dunia industri.
 - b. Sebagai bentuk kontribusi terhadap masyarakat dalam mewujudkan pengembangan teknologi.

G. Keaslian Gagasan

Berikut ini beberapa penelitian yang relevan, yang bisa dijadikan acuan untuk Karya proyek akhir ini diantaranya sebagai berikut :

1. Jurnal : Sistem Akuisisi EKG Menggunakan USB Untuk Deteksi Aritmia karya Arif Widodo dari Fakultas Teknologi Industri Institut Teknologi Sepuluh November di Surabaya tahun 2010. Pada penelitian ini dengan menggunakan rangkaian penguat instrumentasi dan Atmega8 sebagai kendali utama, komunikasi serial RS 232 dan modul USB to serial yang bertujuan untuk menampilkan gelombang elektrokardiograf jantung yang nantinya bisa di tampilkan pada komputer melalui koneksi kabel USB ini masih belum cukup praktis.
2. Jurnal : Rancang Bangun Elektrokardiograf (EKG) karya Tyas Istiqomah dari Fakultas Sains Universitas AIRLANGGA Surabaya tahun 2012. Pada penelitian ini EKG yang dibuat menggunakan rangkaian instrumentasi untuk mendeteksi sinyal jantung, seperti catu daya, diferensial amplifier,

buffer, rangkaian filter dan IC LM 324, penggunaan rangkaian instrumentasi ini dirasa terlalu rumit untuk hanya sekedar menampilkan gelombang jantung di monitor.

Berdasarkan penelitian sebelumnya alat yang digunakan untuk menampilkan gelombang aktifitas kelistrikan jantung masi tarlalu rumit karena masih menggunakan berbagai macam rangkaian penguat dan juga filter instrumentasi pendukung untuk menampilkan hasil gelombang elektrokardiografi jantung. Dalam hal ini penulis ingin meyediakan suatu alat pemantau gelombang elektrokardiografi jantung yang mampu menampilkan gelombang PQRSJT jantung secara jelas dengan menggunakan sensor elektroda dan modul EKG shield standar pemantau gelombang listrik jantung manusia dan penggunaan Arduino Mega 2560 sebagai kendali utamanya.

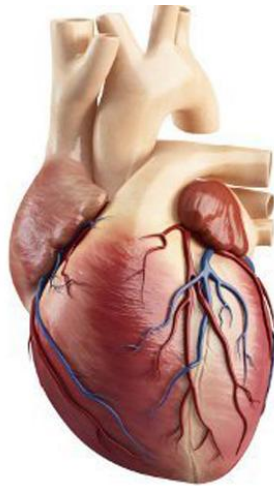
Berdasarkan hal diatas, karya ini benar-benar gagasan serta rancangan dari penulis. Jika ada kesamaan dalam beberapa hal tapi penulis yakin ada perbedaan mendasar baik secara teknik dan konsep alat yang diusung.

BAB II

PENDEKATAN PEMECAHAN MASALAH

A. Jantung

Jantung adalah salah satu organ tubuh yang paling vital fungsinya dibandingkan dengan organ tubuh lainnya jika dilihat secara fisiologis. Adapun pengertian lain dari jantung yaitu organ berotot yang berkontraksi secara ritmis, memompa darah melalui sistem sirkulasi (MEP Books, 2016). Apabila fungsi jantung mengalami gangguan maka hal ini akan berpengaruh pada organ tubuh lainnya, terutama pada ginjal dan otak karena fungsi utama jantung adalah untuk mensirkulasikan darah ke seluruh tubuh sebagai metabolisme sel agar tetap berkembang.



Gambar 1. Jantung Manusia

(Sumber : Bramardianto.com)

Jantung terdiri dari dua bagian, yaitu jantung bagian kanan dengan fungsi sebagai pemompa darah ke paru-paru dan jantung bagian kiri sebagai pemompa darah keseluruh tubuh. Pada masing-masing bagian jantung terdapat dua bilik

(ruang) yaitu atrium pada bagian atas jantung dan ventrikel pada bagian bawah jantung. Masing-masing atrium adalah sebuah pompa yang lemah untuk memompa darah ke ventrikel. Ventrikel sebagai pompa utama pengiriman darah baik ke paru-paru melalui ventrikel kanan dan seluruh tubuh melalui ventrikel kiri. Ketika sistem ini berfungsi normal, atrium berkontraksi kira-kira seper enam detik mendahului kontraksi ventrikel, sehingga memungkinkan pengisian ventrikel sebelum ventrikel memompa darah menuju paru-paru dan tubuh.

Pada jantung terdapat otot yang mirip dengan otot rangka karena mekanisme kontraksi keduanya hampir sama. Namun terdapat beberapa perbedaan pada keduanya, yaitu otot jantung bekerja tanpa perintah neuron seperti pada otot rangka karena kerja otot jantung adalah otomatis. Perbedaan kedua terletak pada durasi kontraksi keduanya yang mana durasi (lama) kontraksi otot jantung lebih lama dibandingkan otot rangka. Jantung tersusun dari beberapa jenis otot, yaitu otot *atrial*, otot *ventricular*, otot *excitatory*, dan serat otot *conductive*. Kontraksi jantung yang kuat terletak pada otot *atrial* dan otot *ventricular*. Otot *excitatory* dan otot *conductive* berkontraksi sangat lemah karena hanya terdiri dari sedikit benang-benang fibril. Namun, masing-masing otot tersebut berkontraksi secara otomatis hingga dihasilkan arus listrik dalam bentuk potensial aksi atau konduksi jantung dan ritme jantung dapat dikontrol (Kurachi, 2001). Sewaktu impuls jantung melewati jantung, arus listrik akan menyebar kedalam jaringan di sekeliling jantung dan sebagian kecil dari arus tersebut akan menyebar ke permukaan tubuh yang lain.

Jantung dapat terserang penyakit sehingga jantung tidak dapat menjalankan tugasnya dengan baik. Hal tersebut antara lain:

1. Otot jantung yang lemah. Ini adalah suatu kelainan bawaan sejak lahir.

Otot jantung yang lemah membuat penderita tak dapat melakukan aktifitas yang berlebih, karena pemaksaan kinerja jantung yang berlebihan akan menimbulkan rasa sakit di bagian dada, dan kadangkala dapat menyebabkan tubuh menjadi nampak kebiru-biruan. Penderita lemah otot jantung ini mudah pingsan.

2. Adanya celah antara serambi kanan dan serambi kiri, oleh karena tidak sempurnanya pembentukan lapisan yang memisahkan antara kedua serambi saat penderita masih dalam kandungan. Hal ini menyebabkan darah bersih dan darah kotor tercampur. Penyakit ini juga membuat penderita tidak dapat melakukan aktifitas yang berat, karena aktifitas yang berat hamper dapat dipastikan akan membuat tubuh penderita menjadi biru dan sesak nafas, walaupun tidka menyebabkan rasa sakit di dada, Ada pula variasi dari penyakit inim yakni penderitanya benar – benar hanya memiliki satu buah serambi (Azhar, 2009).

B. Elektrofisiologis Sel Jantung

Dinding sel dalam tubuh manusia pada umumnya merupakan membran semipermeabel yang hanya dapat melewatkan zat-zat tertentu (Rosyadi, 2001). Sel-sel dalam tubuh tersusun dari protoplasma yang mengandung ion-ion yang terjadi akibat proses ionisasi. Ion-ion yang dominan adalah Na^+ (sodium), K^+

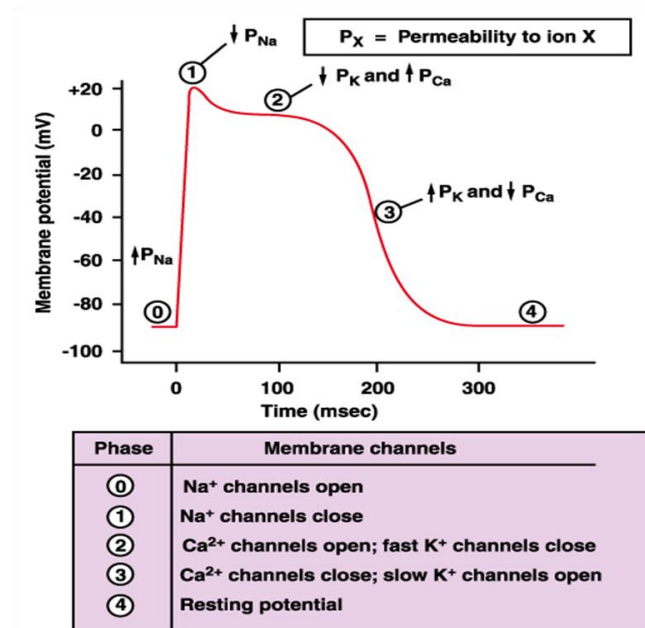
(potassium), dan Cl^- (klorida). Terdapat beribu-ribu kanal ion pada membran sel-sel otot jantung (myocardium) yang merupakan jalur utama bagi ion-ion untuk berdifusi. Kanal-kanal tersebut bersifat relatif spesifik terhadap ion-ion tertentu, misalnya kanal Kalsium dilalui Ca^{++} , kanal Kalium dilalui K^+ , kanal Natrium dilalui Na^+ , dan seterusnya. Selain itu, kanal-kanal ion tersebut dikontrol oleh suatu mekanisme “pintu gerbang” sehingga dapat membuka dan menutup tergantung pada kondisi trans membrane.

Ion-ion cenderung membentuk persamaan elektron didalam dan diluar sel, sehingga terjadi distribusi yang tidak seimbang dan menimbulkan suatu gaya Tarik menarik antara ion-ion dimana ion negatif (terutama anion organik) berkumpul dipermukaan dalam, sedangkan ion positif (terutama Na^+) berkumpul di permukaan luar membrane sel. Keadaan ini dikatakan sel berada dalam stadium polarisasi.

Ion-ion memiliki muatan listrik dan pada waktu sel tidak aktif, terdapat perbedaan potensial antara permukaan dalam dan luar membran sel sebesar 90 mV, dimana muatan intra sel lebih negatif dibandingkan muatan ekstra sel sehingga ditulis -90 mV. Keadaan ini terjadi saat membran sedang istirahat dan keadaan demikian dikatakan bahwa membran mengalami polarisasi. Membran yang sedang istirahat disebut dengan keadaan tanpa rangsang atau *resting state*.

Pada keadaan ada rangsang terjadi kenaikan potensial membran plasma disebut depolarisasi dimana terjadi kenaikan potensial membran sebesar +35mV. Saat depolarisasi terjadi peningkatan ion Na^+ ekstra sel yang menyebabkan permeabilitas membran terhadap ion Na juga meningkat sehingga

dapat masuk ke intra sel dan intra sel bermuatan positif. Bila rangsang sudah tidak ada lagi terjadi penurunan potensial membran. Membran yang mengalami hiperpolarisasi mempunyai negatif potensial membran lebih besar dari pada saat potensial istirahat atau potensial membran depolarisasi. Pada saat hiperpolarisasi terjadi perpindahan ion Kalium dari intrasel ke luar menuju ekstra sel sehingga intrasel bermuatan negatif. Selanjutnya membran mengalami proses repolarisasi yaitu proses pengembalian ion-ion Na^+ ke luar sel dimana membran kembali ke potensial istirahat. Keseluruhan proses tersebut disebut dengan potensial aksi yang dapat dilihat pada gambar dibawah.

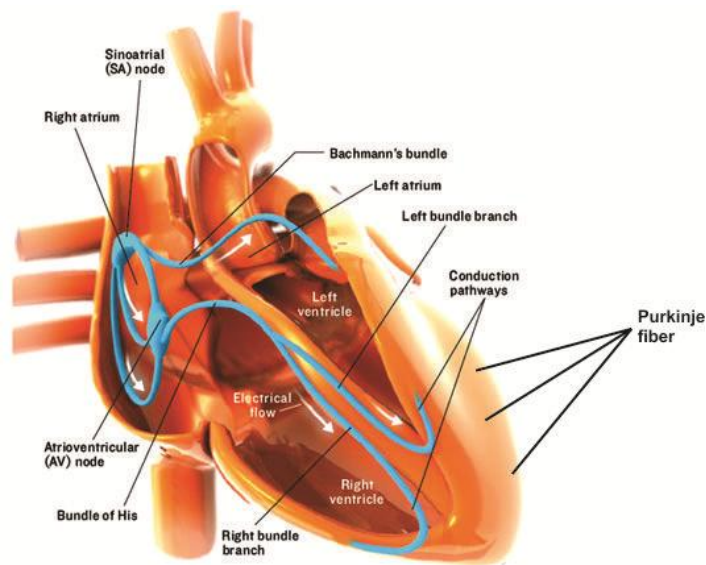


Gambar 2. Potensial Aksi Jantung

(Sumber: <https://www.scribd.com/doc/219215722/Jantung-Pemompa-Darah>)

C. Konduksi Jantung

Kontraksi otot manapun akan selalu menimbulkan perubahan kelistrikan yang dikenal dengan istilah potensial aksi. Potensial aksi bisa terjadi bila suatu daerah membran syaraf atau otot mendapat rangsangan daerah sekitar sel membran untuk mencapai nilai ambang. Potensial aksi yang timbul pada otot jantung (miokardium) dan jaringan transmisi jantung inilah yang memberikan gambaran kelistrikan jantung (konduksi jantung). Adanya konduksi jantung dapat menghasilkan impuls listrik secara ritmis yang menyebabkan adanya kontraksi ritmis otot jantung yang disebut ritme jantung, mengirim potensial aksi melalui otot jantung dan menyebabkan terjadinya detak jantung (Guyton, 2006)



Gambar 3. Struktur Konduksi Jantung

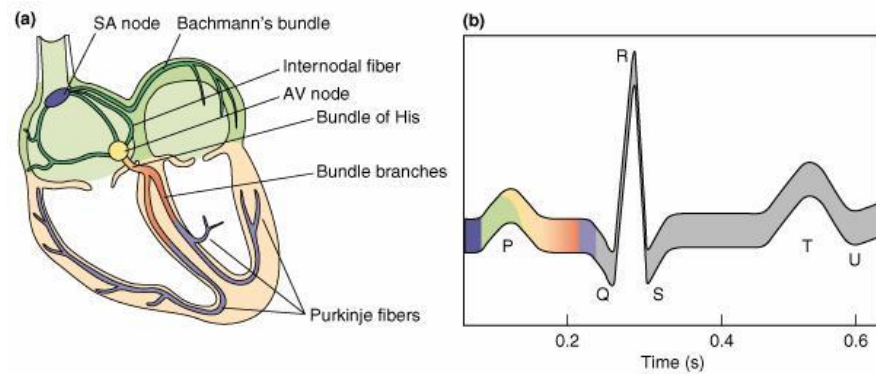
(Sumber: <http://elmanbillonx.blogspot.co.id/2012/06/konduksi-jantung.html>)

Arah konduksi jantung berdasarkan Gambar 3 adalah dari simpul sinoatrial (SA) yang terletak pada bagian atas serambi kanan. Simpul SA inilah yang menimbulkan rangsangan yang menyebabkan jantung berkontraksi. Simpul atrioventrikular (AV) terletak pada dinding yang membatasi serambi kanan dan bilik kanan. Simpul ini berfungsi menghantarkan impuls dari serambi ke bilik. Impuls dari simpul AV kemudian diteruskan ke seluruh bilik melalui berkas His. Pada ujung berkas His terdapat banyak cabang. Cabang-cabang ini disebut serat Purkinje. Serat Purkinje bertugas meneruskan impuls dari berkas His ke seluruh otot bilik. Bilik kemudian berkontraksi sehingga darah dipompa keluar dari bilik dan mengalir dalam sistem peredaran darah (Jones, 2005).

Konduksi jantung berhubungan dengan jumlah *heart rate* (detak jantung) per menit. *Heart rate* digunakan sebagai indikasi awal dalam menentukan ada atau tidaknya kelainan pada jantung. Jumlah normal *heart rate* adalah 60-100 kali/menit. Jika jumlah *heart rate* dibawah 60 kali/menit maka terjadi bradikadi, sedangkan jumlah *heart rate* yang di atas 100 kali/menit terjadi takikardi.

Perbedaan potensial membrane istirahat pada sistem konduksi jantung yang terdiri atas simpul SA, atrium simpul AV, His-purkinje, dan ventrikel bergantung pada tipe potensial aksi masing-masing. Potensial aksi ini teridiri atas 2 tipe berdasarkan penyebab depolarisasi primer, yaitu tipe potensial aksi respon cepat dan potensial aksi respon lambat. Atrium, his-purkinje dan ventrikel memiliki karakteristik potensial aksi respon yang cepat. Sedangkan yang memiliki karakter potensial aksi respons lambat yaitu simpul SA dan

simpul AV. Pada gambar dibawah menunjukan potensial membrane yang terjadi pada pemicu jantung.

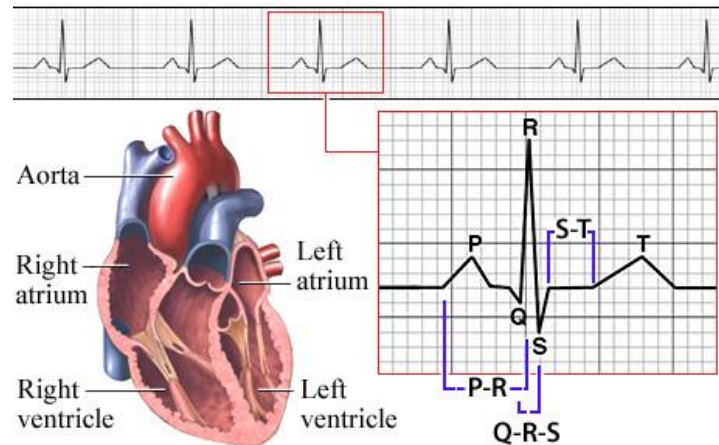


Gambar 4. Potensial Membran Pemicu Jantung

(Sumber : <http://instrumentasi.lecture.ub.ac.id/tag/jantung/>)

D. EKG (Elektrokardiograf)

EKG adalah suatu gambaran grafis mengenai gambaran puncak aktifitas listrik dari serabut otot jantung, berupa kurva tegangan fungsi waktu yang terdiri dari berbagai puncak (Heru, 2008). Sebuah EKG dapat digunakan untuk mengukur denyut jantung, mendiagnosis adanya infark mikroad yang sedang berkembang, mengidentifikasi aritmia dan efek dari obat dan peralatan yang digunakan pada penanganan jantung.



Gambar 5. Sinyal keluaran EKG

(Sumber : <http://ikabeliapratiwi.blogspot.co.id>)

Proses terjadinya sinyal elektrokardiograf tersebut adalah sebagai berikut :

1. Gelombang P berasal dari kontraksi atrium dari sinus atrialis ke nodus atrio ventrikular saat darah mulai memasuki jantung dari seluruh tubuh.
2. Gelombang R adalah tanda akhir dari kontraksi atrium dan awal dari kontraksi ventrikel saat memasuki ruang ventrikel.
3. Kompleks QRS berasal dari adanya aktivitas kontraksi pada ventrikel yang memompakan darah ke seluruh tubuh dan merupakan gelombang tertinggi.
4. Gelombang T berasal dari repolarisasi ventrikel atau ventrikel kembali dalam keadaan istirahat setelah proses pemompaan darah ke seluruh tubuh selesai.
5. Interval P-R menandakan waktu dari permulaan kontraksi atrial sampai permukaan kontraksi ventrikel.

6. Interval R-T menunjukkan kontraksi otot (ventrikel sistole), dan interval T-R menyebabkan adanya relaksasi otot (ventrikel diastole). (Hadiyoso, 2011)

Selain itu EKG memiliki cara kerja untuk mengukur aktifitas kelistrikan yang dihasilkan jantung pada tiap denyut jantung. Proses ini bisa memantau kelainan yang mungkin ada seperti aritmia jantung, peradangan, sampai jantung coroner. Berikut ini adalah fungsi EKG secara spesifik :

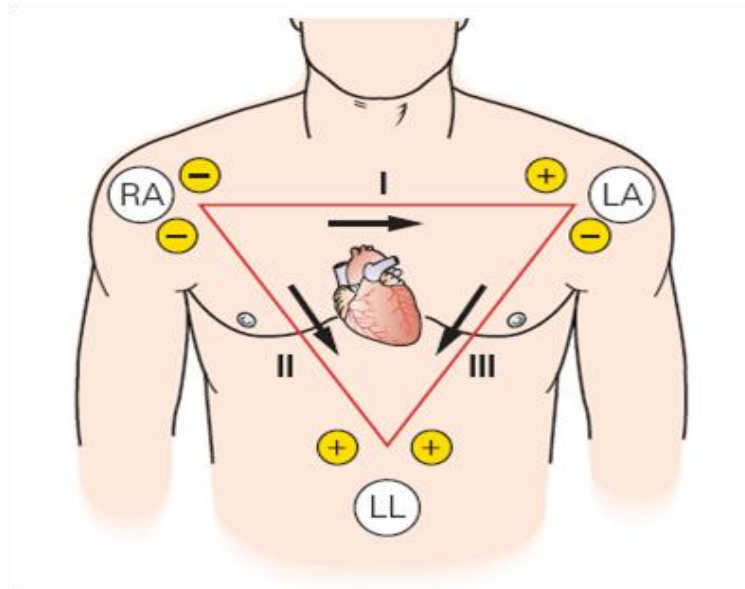
1. Mengetahui denyut dan irama jantung yang bisa mengindikasikan adanya kelainan jantung.
2. Mengetahui posisi jantung dalam rongga dada. Posisi normal jantung seseorang berada di area dada kiri, dipeluk oleh paru-paru sebelah kiri.
3. Mengetahui ada atau tidaknya penebalan otot jantung atau hipertrofi. Hipertrofi juga dikenal sebagai kondisi massa otot jantung yang bertambah, sehingga juga terjadi penebalan atau pembengkakan. Hal ini akan membuat jantung meningkatkan curah (jumlah darah yang dipompa keluar dari ventrikel kiri tiap menit).
4. EKG jantung utamanya bisa berfungsi dalam mengindikasikan adanya pola aktifitas listrik jantung yang mungkin menyebabkan gangguan irama atau aritmia.
5. Mendeteksi sedini mungkin penyakit jantung *Ischema*. Jantung *Ischema* adalah kondisi tidak tercukupinya *supply* darah ke jaringan dan organ-organ tubuh berakibat permasalahan pada pembuluh darah, terutama di

area jantung. Masalah penyebabnya bisa jadi karena penumpukan kolesterol pada pembuluh darah.

6. Mendeteksi sedini mungkin penyakit *Myocardiac Infraction*. *Myocardiac Infraction* dikenal dengan nama lain serangan jantung, yaitu sebuah kondisi terhentinya aliran darah meskipun hanya sesaat dari jaringan ke seluruh tubuh, sehingga mengakibatkan sebagian sel jantung mati. Penyebab yang kerap kali dituding sebagai biang kerok *Myocardiac Infraction* adalah penyumbatan pembuluh darah.
7. Mendeteksi adanya efek obat-obatan yang dikonsumsi seseorang. EKG berfungsi menemukan adanya kelainan yang mungkin terjadi dari konsumsi obat-obatan tertentu yang berpengaruh pada kinerja jantung.

E. Titik Sadapan Sinyal (lead) EKG

Penggunaan EKG dilengkapi dengan pemasangan sadapan pada tubuh sebagai monitor adanya perubahan tegangan antara elektroda yang ditempatkan pada berbagai posisi di tubuh. Pengukuran sinyal pada EKG dilakukan dengan pemilihan tiga titik bipolar yang pertama kali dikenalkan oleh *Einthoven* yang terletak di lengan kanan, lengan kiri, dan kaki kiri. Pengambilan titik *refence* ini kemudian di kenal dengan segitiga *Einthoven* yang digambarkan pada gambar 6.



Gambar 6. Titik Referensi Pemasangan Elektroda 3 Lead

(Sumber : <http://dokter-medis.blogspot.co.id/2009/07/elektrokardiografi-ekg.html>)

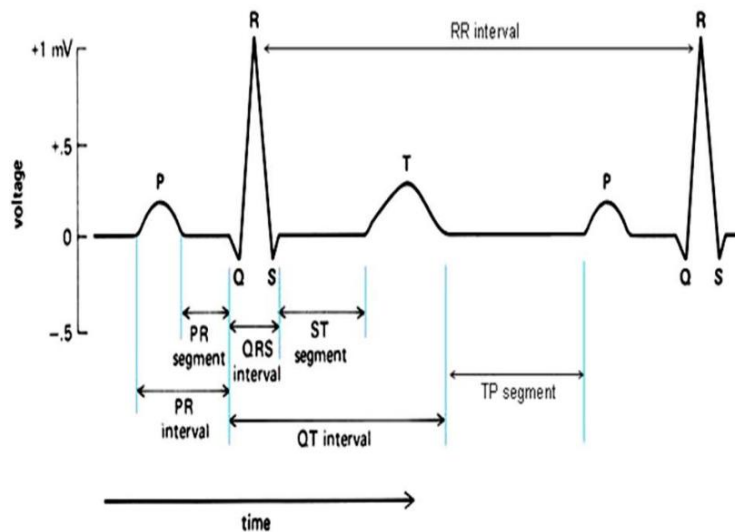
Setelah dilakukan penentuan titik dengan menggunakan prinsip segitiga *Einthoven* selanjutnya adalah melakukan proses penempelan elektroda yang merupakan pencacatan sinyal elektrik jantung dari gabungan lead yang ditempatkan pada titik-titik tersebut.

F. Pembacaan EKG

Mesin EKG merupakan alat yang digunakan untuk mengolah sinyal elektrik jantung melalui elektroda dan menampilkannya lewat kertas/layar monitor dengan skala tertentu. Aksis horizontal mewakili waktu dengan kecepatan 25mm/detik. Setiap kotak kecil kertas EKG berukuran 1 mm². Dengan kecepatan 25 mm/s, 1 kotak kecil kertas EKG sama dengan 0,04 s (40 ms). Lima kotak kecil menyusun 1 kotak besar, yang sama dengan 0,20 s (200ms) yang mana ada 5 kotak besar per menit. Standarisasi untuk voltase (amplitude) adalah 1, artinya

10 kotak kecil vertikal (1 cm) mewakili 1 mV. Harus menggerakkan jarum 1 cm secara vertical, yakni 2 kotak besar di kertas EKG. Kalibrasi 1 mV diperlukan untuk standar perbandingan antara besarnya input dengan besarnya penguatan amplifier sesuai dengan kepekaan (*sensitivity*) yang dipilih.

Standarisasi ini harus selalu konsisten agar dengan melihat amplitude gambaran EKG, dapat diketahui ada tidaknya perubahan voltase dari konduksi jantung. Hasil dari EKG terdiri atas dua unsur yaitu kompleks dan interval. Komplek yang normal adalah gelombang P, kompleks QRS dan gelombang T (Soeliadi, 1991). Gambar berikut menunjukkan hasil sinyal EKG normal seseorang.



Gambar 7. Sinyal EKG

(Sumber : <https://www.mathworks.com/help/dsp/examples/real-time-ecg-qrs-detection.html>)

Pengukuran EKG dilakukan dengan menggunakan penguat diferensial. Elektroda dua sadapan pada suatu waktu yang dipilih digunakan untuk input ke tahapan penguat diferensial EKG amplifier biasanya memiliki penguatan sekitar

1000x sehingga meningkatkan nilai biopotensial sebesar 1 mV menjadi 1 V. Grafik kecepatan dalam rekaman EKG telah distandarkan pada 25 dan 50 mm/s (McGraw-Hill, 2004).

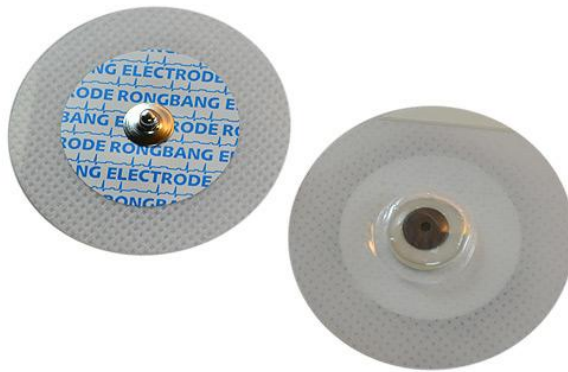
G. Elektroda

Elektroda dibuat bahan yang menjamin resistensi yang rendah antara kulit dan permukaan elektroda. Fungsi utama elektroda adalah untuk mendeteksi sinyal listrik jantung yang merambat melalui bagian tubuh. Tujuannya yaitu mengkonversi informasi biologis menjadi sinyal elektrik yang dapat terukur. Selain itu elektroda untuk mengambil sinyal biolistrik berdasarkan prinsip bahwa kontak antara ion metal dengan metal yang bersesuaian menghasilkan potensial listrik yang disebut potensial elektroda. Potensial elektroda dihasilkan dari material yang memiliki resistansi rendah antara kulit dan permukaan elektroda. Menurut polaritasnya, maka elektroda EKG dapat dibagi menjadi elektroda positif (anoda) , negative (katoda) dan netral (*ground electrode*).



Gambar 8. *Electrode snap cable*

(Sumber : <https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG-PRO/>)

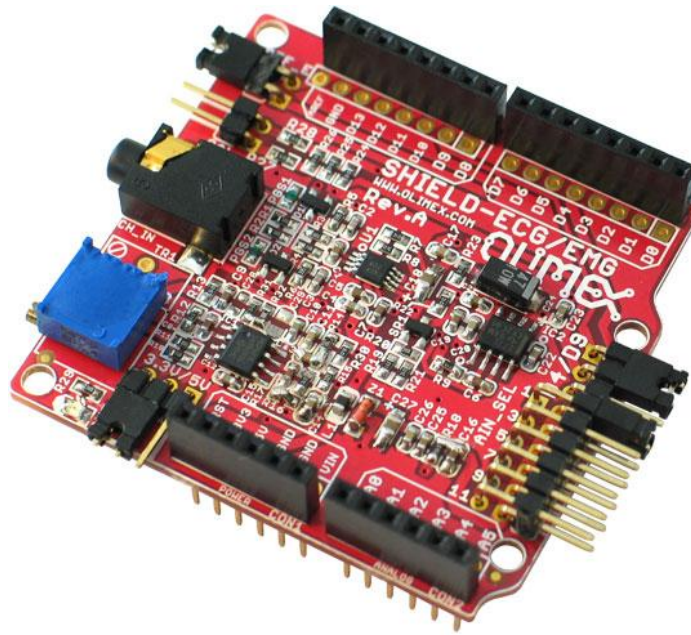


Gambar 9. *EKG Gel Electrodes*

(Sumber : <https://www.olimex.com/Products/Modules/Biofeedback/EKG-GEL-ELECTRODE/>)

H. EKG Shield Olimex

EKG shield adalah sebuah modul sensor yang memberikan akses kepada arduino *boards* untuk menangkap sinyal elektrokardiografi. EKG Shield membuka kemungkinan baru suatu *experiment* dengan menggunakan *biofeedback*, sehingga dengan menggunakan EKG shield tersebut *user* mampu memonitor detak jantung dan mencatat gelombang jantung, mengenali gerak melalui monitoring dan analisa pergerakan otot pada jantung.



Gambar 10. EKG Shield Olimex

(Sumber : <https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/>)

EKG shield mengubah sinyal diferensial analog (EKG bio potensial yang dihasilkan oleh otot), terpasang pada input CH1_IN+/CH1_IN- menjadi sebuah aliran data tunggal sebagai output. Output sinyal tersebut analog dan harus di diskretisasi lebih jauh dengan tujuan memberikan pilihan pemrosesan digital.

I. Arduino

Arduino adalah sebuah *platform* dari *physical computing* yang bersifat *open source* (Aslamia, 2015). Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi merupakan kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih.

IDE adalah sebuah *software* yang sangat berperan yang digunakan untuk menulis program, meng-*compile* menjadi kode biner dan mengunggahnya ke dalam memori mikrokontroler. Menurut (Syahwil, 2013), Arduino adalah papan elektronik *open source* yang di dalamnya terdapat komponen utama, yaitu sebuah *chip* mikrokontroler AVR dari perusahaan Atmel. Berikut ini karakteristik dan struktur arduino sebagai berikut:

1. Arduino IDE merupakan multi *platform* yang dapat dijalankan pada berbagai sistem operasi seperti *Windows* ataupun *Linux*. IDE adalah program komputer yang berfungsi untuk menyediakan semua fasilitas yang diperlukan dalam pembangunan perangkat lunak baik *editor*, *compiler*, *linker*, maupun *debugger*.
2. Pemrograman pada *hardware* Arduino menggunakan kabel yang terhubung dengan *port Universal Serial Bus* (USB). Hal ini disebabkan karena banyak komputer sekarang yang tidak dilengkapi dengan *port* serial.
3. Arduino adalah *hardware* dan *software* yang bersifat *open source* yaitu sistem pengembangan yang tidak dikoordinasi oleh individu atau lembaga pusat, namun oleh para pelaku yang bekerja sama dengan memanfaatkan kode sumber (*source code*).

Biaya pembelian *hardware* tergolong cukup murah sehingga tidak akan menghabiskan banyak uang untuk memilikinya.

1. Arduino Mega 2560

Arduino Mega 2560 adalah papan pengembangan mikrokontroller yang berbasis arduino dengan menggunakan chip ATmega2560. Board ini memiliki pin I/O yang cukup banyak, sejumlah 54 buah digital I/O pin (15 pin diantaranya adalah

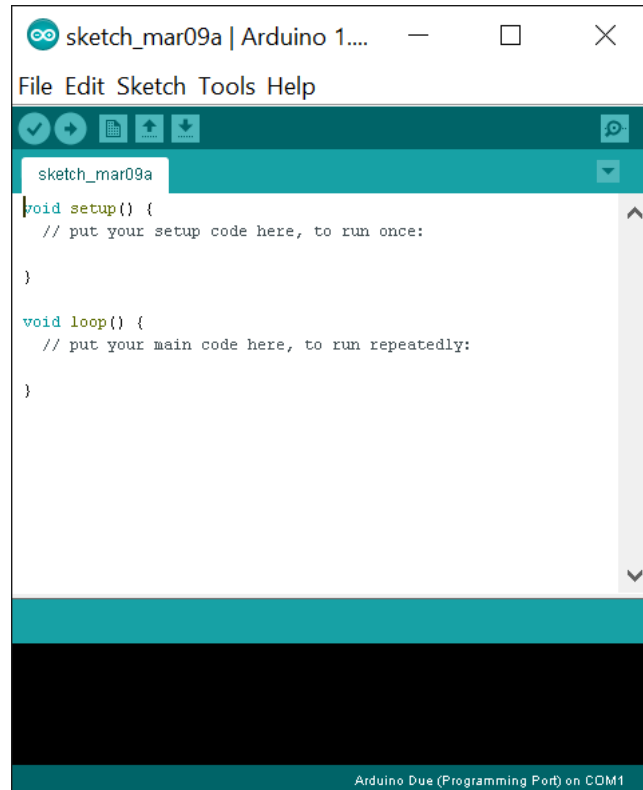
PWM), 16 pin analog input, 4 pin UART (serial port hardware). Arduino Mega 2560 dilengkapi dengan sebuah oscillator 16 Mhz, sebuah port USB power Jack DC, ICSP header, dan tombol reset. Board ini sudah sangat lengkap, sudah memiliki segala sesuatu yang dibutuhkan untuk sebuah mikrokontroler.



Gambar 11. Arduino Mega 2560

2. Arduino IDE

Arduino IDE adalah *software* yang dirancang menggunakan bahasa pemrograman Java yang terdiri dari 3 bagian yaitu *Editor* program, *Compiler*, dan *Uploader*. Pada bagian *editor*, *user* dapat melakukan pemrograman baik itu menulis ataupun mengedit program dalam bahasa *processing*. Sementara pada bagian *Compiler* berisi pengubah kode program menjadi kode biner agar kode tersebut dapat terbaca oleh mikrokontroler. Sedangkan bagian *Uploader* digunakan untuk menuliskan dan menyalin kode biner dari komputer ke dalam memori *board* Arduino. Bahasa pemrograman yang digunakan pada Arduino IDE yaitu jenis bahasa pemrograman C yang mudah digunakan.



Gambar 12. Tampilan Arduino IDE

Pada tampilan Arduino IDE terdapat *toolbar* yang didesain untuk mempermudah dalam melakukan pemrograman. Berikut ini fungsi-fungsi pada *toolbar IDE* sebagai berikut:

- a. *Verify*, digunakan untuk melakukan kompilasi program yang saat di *editor*.
- b. *New*, digunakan untuk membuat program baru dengan mengosongkan isi dari jendela *editor* saat ini.
- c. *Open*, digunakan untuk membuka program yang ada dari sistem file.
- d. *Save*, digunakan untuk menyimpan program saat ini.
- e. *Upload*, digunakan untuk menyalin data hasil pemrograman dari komputer ke dalam *memory board* arduino. Ketika melakukan *upload*, maka harus melakukan pengaturan jenis arduino dan *port* COM yang digunakan.

- f. *Serial Monitor*, digunakan untuk melihat hasil pemrograman yang telah tersimpan dalam *memory* arduino.

Kode program yang dituliskan pada *chip* mikrokontroler Arduino umumnya menggunakan beberapa fungsi seperti tipe data, operator, dan program kontrol. Pada setiap bagian dari data yang disimpan dalam program Arduino umumnya memiliki tipe datanya masing-masing. Berikut ini beberapa tipe data yang terdapat pada Arduino yang ditunjukkan pada Tabel 1.

Tabel 1. Tipe Data Bahasa C

JENIS	RANGE
<i>int / sign int</i>	-32768 - +32767 ($2^{15} - 1$)
<i>unsign int</i>	0 – 65535
<i>short int / signed short int</i>	-28 – 127
<i>unsigned short int</i>	0 – 255
<i>long int / signed long int</i>	-2147483648 – 2147483648
<i>unsigned long int</i>	0 – 4294967296
<i>Char</i>	karakter ASCII
<i>unsigned char</i>	0 – 255
<i>signed char</i>	-128 – 127
<i>Float</i>	maksimum nilai 6 digit
<i>Double</i>	maksimum nilai 12 digit
<i>long double</i>	maksimum nilai 24 digit

(Widodo, 2016)

Sementara operator merupakan simbol atau karakter yang biasa dilibatkan dalam program untuk melakukan sesuatu operasi atau manipulasi, seperti menjumlahkan dua buah nilai, memberikan nilai ke suatu variabel, membandingkan kesamaan dua buah nilai. Sebagian operator C tergolong sebagai operator binari, yaitu operator yang dikenakan terhadap dua buah nilai (*operand*). Berikut ini beberapa operasi kondisi yang sering digunakan dalam bahasa C yang dapat dilihat pada Tabel 2 sebagai berikut:

Tabel 2. Operasi Kondisi

Operator	Keterangan
<	Lebih kecil
<=	Lebih kecil atau sama dengan
>	Lebih besar
>=	Lebih besar atau sama dengan
=	Sama dengan
!=	Tidak sama dengan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Sisa Bagi (modulus)
!	<i>Boolean NOT</i>
&&	<i>Boolean AND</i>
	<i>Boolean OR</i>
~	Komplemen <i>Bitwise</i>
&	<i>Bitwise AND</i>
	<i>Bitwise OR</i>
^	<i>Bitwise Exclusive OR</i>
>>	<i>Right Shift</i>
<<	<i>Left Shift</i>
=	Untuk memasukan nilai
+=	Untuk menambah nilai dari keadaan semula
-=	Untuk mengurangi nilai dari keadaan semula
*=	Untuk mengalikan nilai dari keadaan semula
/=	Untuk melakukan pembagian terhadap bilangan semula
%=	Untuk memasukan nilai sisa bagi dari pembagian bilangan semula
<<=	Untuk memasukan <i>shift left</i>
>>=	Untuk memasukan <i>shift right</i>
&=	Untuk memasukan <i>bitwise AND</i>
^=	Untuk memasukan <i>bitwise XOR</i>
\ =	Untuk memasukan <i>bitwise OR</i>

(Widodo, 2016)

Selain itu, dalam melakukan pemrograman menggunakan Arduino IDE juga dibutuhkan program kontrol agar program yang akan dibuat sesuai dengan tujuan.

Berikut ini program kontrol dalam bahasa C sebagai berikut:

1. Percabangan *if* dan *if ... else ...*

Perintah *if* dan *if ... else ...* digunakan untuk melakukan operasi percabangan bersyarat.

2. Percabangan *switch*

Pernyataan *switch* adalah sebuah variabel secara berurutan diuji oleh beberapa konstanta bilangan bulat atau konstanta karakter sintaks perintah *switch*.

3. *Looping*

Looping adalah pengulangan satu atau beberapa perintah sampai mencapai keadaan tertentu. Ada tiga perintah *looping*, yaitu:

a). *for ...*

b) *while ...*

c) *do...while....*

J. FreeHC

FreeHC adalah suatu program komputer untuk neurofeedback. Meskipun program ini masih dalam pengembangan tahap awal, sehingga versi yang ditawarkan masih kasar dan masih perlu penyempurnaan. Fitur pada versi saat ini program masih menggunakan transformasi fourier cepat untuk mengekstrak pita frekuensi dari sinyal mentah. Simulator dibangun dalam program untuk memudahkan verifikasi. Dapat diperhatikan pada saat program FreeHC digunakan untuk menampilkan gelombang ekg pada program ini bukan hanya menampilkan gelombang jantung saja, tetapi juga banyak heart rate (bpm).



Gambar 13. Program FreeHC

BAB III

KONSEP RANCANGAN

A. Analisa

Analisa Rancang Bangun Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield, memerlukan perincian kebutuhan alat dan bahan untuk memudahkan proses pembuatan alat yang meliputi identifikasi dan analisa kebutuhan.

1. Identifikasi Kebutuhan

Pengidentifikasian kebutuhan dapat dikelompokkan menjadi 4 bagian, meliputi:

a. Hardware

1. Catu Daya

Catu daya yang digunakan bersumber dari PC/laptop melalui kabel USB untuk penunjang kerja alat

2. Input

Elektroda yang berfungsi sebagai sensor yang berfungsi untuk menyadap sinyal elektrik yang dihasilkan oleh aktivitas kelistrikan jantung.

3. Proses

Mikrokontroler Arduino Mega 2560 sebagai mini sistem untuk mengerjakan atau mengoperasikan *input* dan *output*

4. *Output*

Output yang dikeluarkan adalah sinyal ADC yang kemudian diolah oleh software pada PC yang kemudian akan ditampilkan dalam wujud gelombang elektrokardiograf.

b. *Software*

1. Arduino IDE Sebagai *software* pemrograman untuk *compiling* program yang akan digunakan pada modul.
2. FreeHC sebagai *software* pemrograman untuk menampilkan gelombang elektrokardiograf di layar monitor.

2. **Analisis Kebutuhan**

Berdasarkan identifikasi kebutuhan di atas, maka perlu dilakukan analisa kebutuhan terhadap pengembangan alat yang akan dirancang agar sistem yang di bangun lebih akurat adalah sebagai berikut:

a. *Hardware*

1. Bagian Sumber Catu Daya

Mebutuhkan tegangan +5V agar modul EKG shield dan arduino dapat beroperasi dengan normal, yang terkoneksi melalui kabel USB pada PC.

2. Bagian *Input*

Alat ini menggunakan sadapan elektroda yang ditempel pada titik yang sudah ditentukan.

3. Bagian Proses

Alat ini membutuhkan komponen yang dapat mengolah data dari *input* yang akan diberikan pada bagian *output*. Arduino Mega 2560 digunakan sebagai sistem kendali yang akan melakukan proses dari data yang diberikan oleh *input* dan akan langsung diproses oleh EKG Shield untuk menghasilkan keluaran sinyal EKG melalui Elektroda yang terpasang sebagai bentuk *output* yang kemudian akan ditampilkan pada program aplikasi yang terbuka dalam PC.

4. Bagian Output

Bagian *output* terdiri dari tampilan *software* pada PC yang akan menampilkan bentuk gelombang EKG.

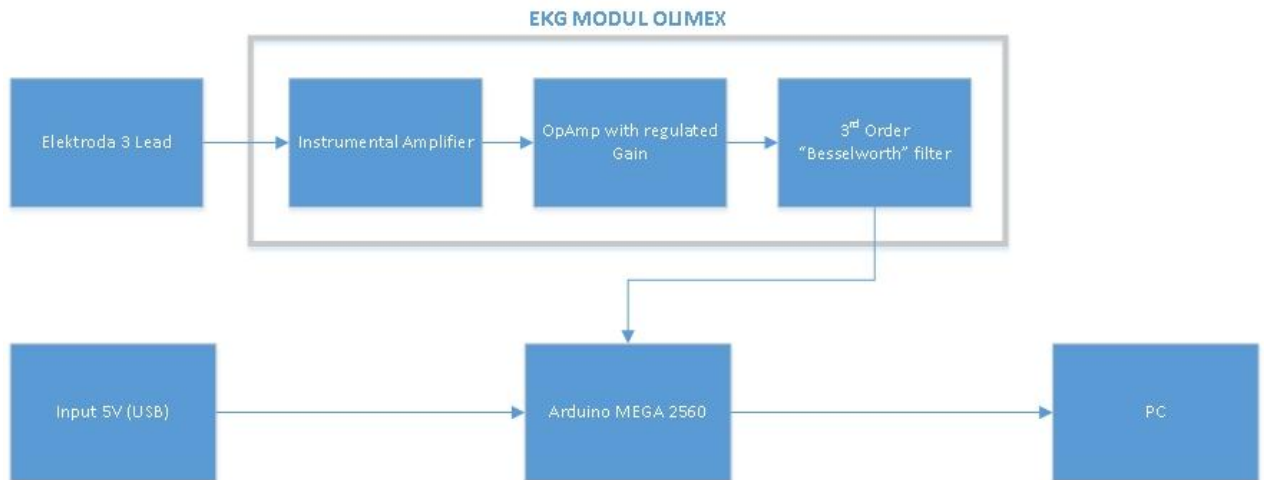
1. Software Arduino IDE

IDE adalah program komputer yang berfungsi untuk menyediakan semua fasilitas yang diperlukan dalam pembangunan perangkat lunak baik *editor*, *compiler*, *linker*, maupun *debugger*. Pemrograman pada *hardware* Arduino menggunakan kabel yang terhubung dengan *port Universal Serial Bus* (USB). Hal ini disebabkan karena banyak komputer sekarang yang tidak dilengkapi dengan *port* serial.

2. FreeHC

FreeHC sangat dibutuhkan untuk melihat hasil simpanan dari alat tersebut yang berupa gelombang jantung.

B. Blok Diagram Sistem



Gambar 14. Blok Diagram Sistem

Gambar 14 merupakan bagian blok diagram rangkaian keseluruhan proses yang diaplikasikan pada Rancang Bangun Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield:

1. Elektroda yang memiliki fungsi sebagai inputan yang akan memberikan tegangan jantung untuk kemudian diolah sinyal nya dalam modul EKG OLIMEX.
2. Sinyal yang ditangkap oleh elektroda, selanjutnya menuju instrumental amplifier, fungsi nya agar impedansi input tetap terjaga dalam posisi tinggi, dan gain yang dihasilkan dapat presisi.
3. Sedangkan OpAmp with regulated gain berfungsi sebagai penguat sinyal dengan gain yang stabil, dengan tegangan offset yang rendah sehingga dapat meminimalisir error pada sisi output nanti.

4. 3rd order Besselworth filter digunakan untuk mengeliminasi *noise* yang terdapat pada rentang gelombang 40Hz
5. Sumber catu daya memiliki fungsi sebagai sumber daya masukan untuk mensuplay arduino mega,dan EKG Shield.
6. Arduino Mega 2560 dan EKG Shield, berfungsi sebagai pengolah data, pengkonversi dan pengendalian sistem
7. PC berfungsi sebagai media penampil untuk hasil gelombang EKG.

Berdasarkan Gambar 14 dapat dijelaskan bagaimana cara kerja dari Rancang Bangun Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield, yaitu dengan menempelkan elektroda pada 3 titik sadapan dengan menggunakan arduino mega 2560 sebagai pengendali dan ekg shield sebagai modul yang nantinya akan menghasilkan gelombang elektrokardiograf yang nantinya akan ditampilkan pada layar PC untuk selanjutnya dianalisa.

C. Perancangan Sistem

Perancangan sistem Rancang Bangun Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield ini terdiri dari Snap kabel elektroda, Arduino Mega 2560, dan EKG Shield olimex :

a. Hardware

1. Rangkaian Arduino Mega 2560

Arduino Mega 2560 pada proyek ini berfungsi sebagai otak yang menjalankan rangkaian yang digunakan. Proyek ini menggunakan beberapa pin dari mikrokontroler diantaranya adalah :

- a. Pin analog A0 hingga A5 difungsikan untuk input analog modul EKG olimex
- b. Pin digital D0 hingga D13 difungsikan untuk input digital modul EKG olimex.
- c. Pin power, digunakan untuk mensuplay tegangan untuk modul EKG Olimex.
- d. Pin IOREF, pin AREF, dan pin Reset difungsikan sebagai control data untuk modul EKG olimex.

2. Rangkaian modul EKG Shield olimex

Modul EKG shield olimex pada proyek ini berfungsi sebagai pemroses input tegangan yang disadap oleh elektroda, untuk diproses agar mendapatkan sinyal EKG. Proyek ini menggunakan elektroda lead yang akan dipasangkan di modul EKG Shield olimex dalam bentuk jack 3,5mm.

b. *Software*

1. Arduino IDE

Software ini dirancang dengan menggunakan bahasa pemrograman Java yang terdiri dari 3 bagian *Editor* program, *compiler*, dan *Uploader*. Pada bagian *editor*, *user* dapat melakukan pemrograman baik itu menulis ataupun mengedit program dalam bahasa *processing*. Sementara pada

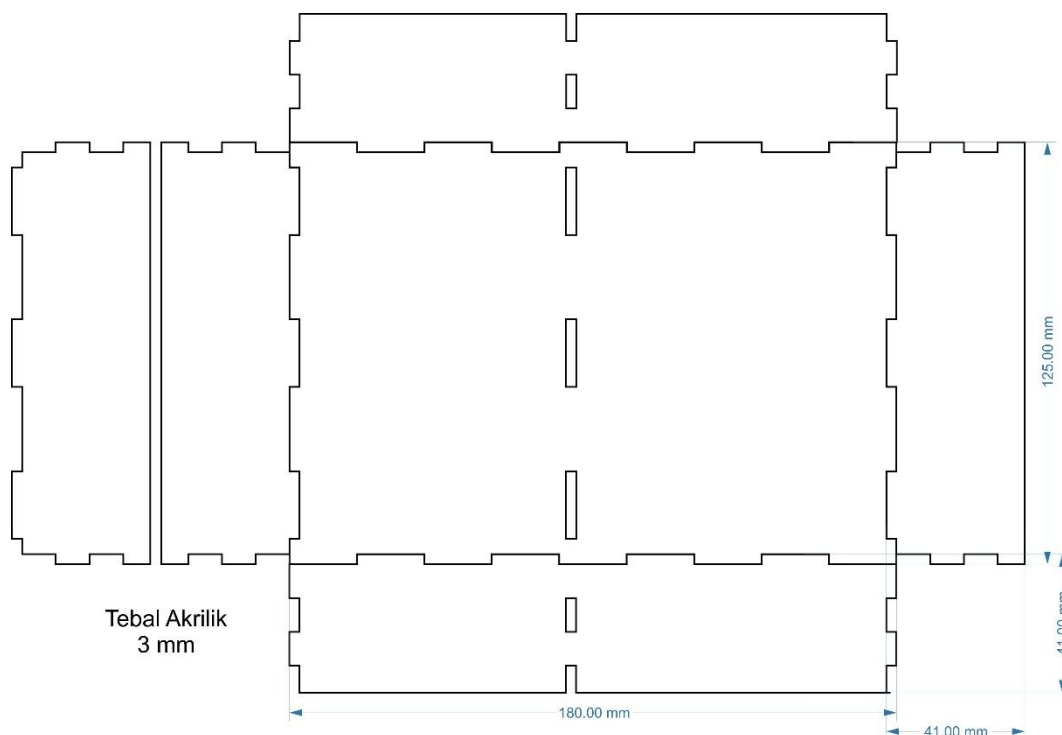
bagian *Compiler* berisi pengubah kode program menjadi kode biner agar kode tersebut dapat terbaca oleh mikrokontroler. Sedangkan bagian *Uploader* digunakan untuk menuliskan dan menyalin kode biner dari komputer ke dalam memori *board* arduino.

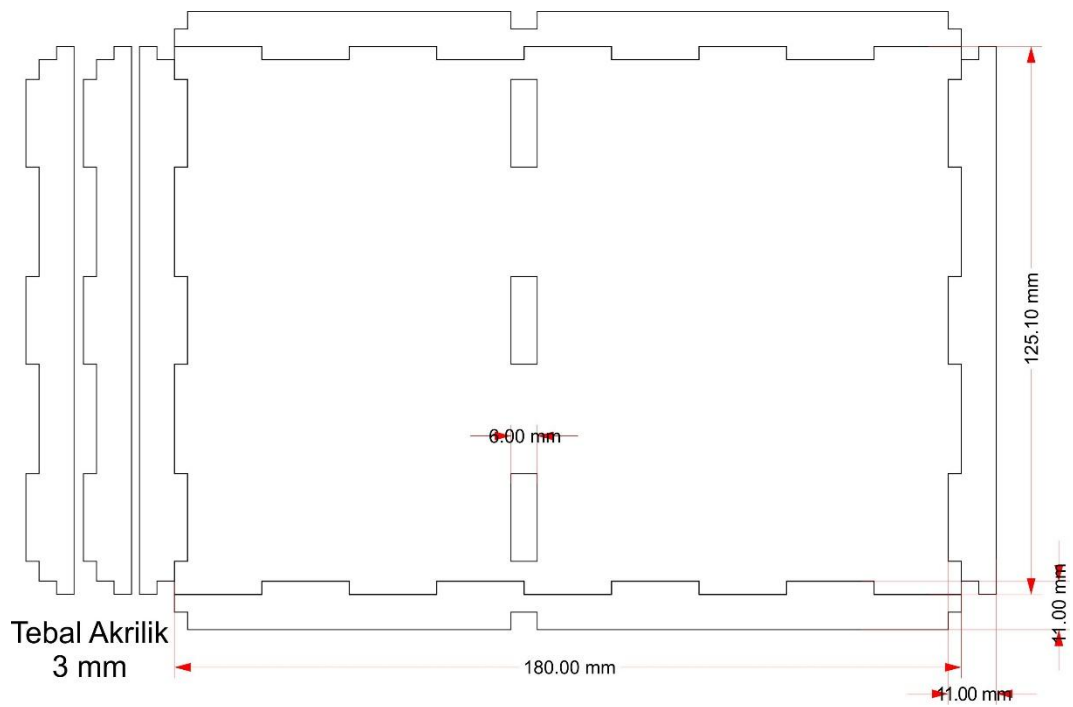
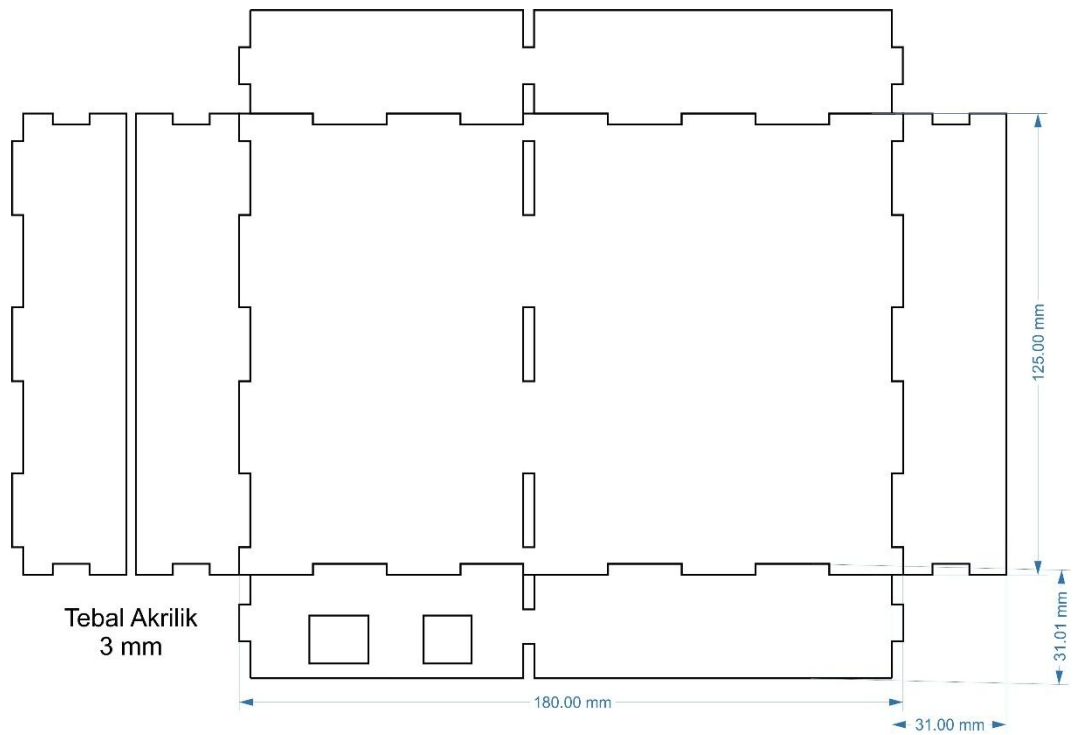
D. Langkah Pembuatan Alat

Langkah pembuatan alat pada proyek akhir ini terdiri dari pembuatan *box*, pembuatan PCB dan pemasangan komponen pada PCB, dan pemasangan rangkaian ke dalam *box*.

1. Pembuatan *box*

Pembuatan *box* menggunakan akrilik sebagai bahan dasarnya dengan desain menggunakan software corelDraw. Gambar 15 merupakan jaringan-jaring pada desain *box* untuk memudahkan pembuatan pada akrilik





Gambar 15. Desain *box*

2. Pemasangan Rangkaian pada Box

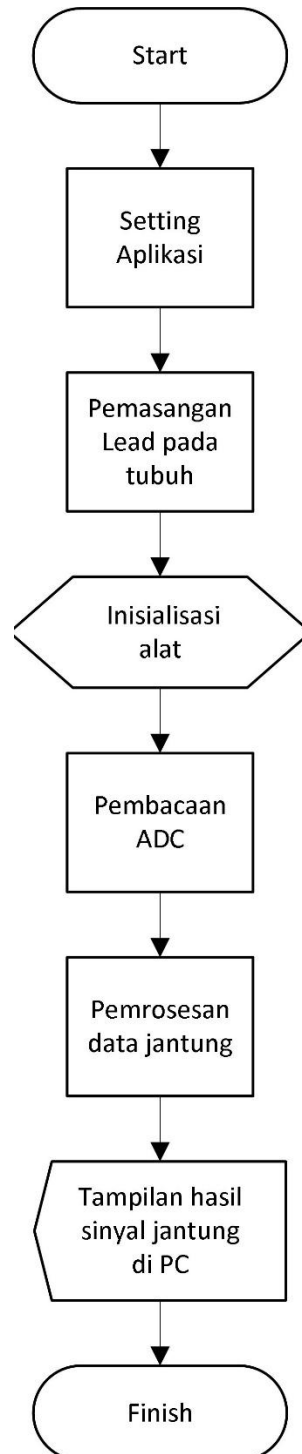
Langkah selanjutnya yaitu menyesuaikan tata letak komponen ke dalam box yang sudah dibuat. Pemasangan disesuaikan dengan desain bagian dalam box.

E. Software

1. Algoritma Program

- a. *Input* elektroda
- b. Inisialisasi Alat
- c. Baca ADC
- d. Penghitung gelombang EKG
- e. Mengirim data ke PC
- f. Menampilkan data di PC
- g. Lakukan pengulangan pembacaan ADC pada tahap 'c'
- h. Finish

2. Flowchart



Gambar 16. Flowchart Sistem

F. Spesifikasi Alat

Rancang Bangun Alat Elektrokardiograf (EKG) Sebagai Pemantau Aktivitas Kelistrikan Jantung Dengan Menggunakan Arduino Mega 2560 dan EKG Shield, mempunyai spesifikasi sebagai berikut :

1. Alat ini menggunakan elektroda dengan snap kabel output berupa jack DC 3,5 mm.
2. Modul EKG Shield olimex, sebagai pengolah data ADC yang diberikan oleh elektroda.
3. Arduino Mega 2560 sebagai otak untuk kendali utama.
4. PC untuk display atau memperagakan gelombang EKG yang diproses.

G. Pengujian Alat

Pengujian alat dilakukan untuk mendapatkan data penelitian. Pengujian alat ini dilakukan dengan dua pengujian, yaitu:

1. Uji Fungsional

Pengujian alat dilakukan dengan cara menguji setiap bagian-bagian berdasarkan karakteristik dan fungsi masing-masing. Pengujian ini dilakukan untuk mengetahui apakah setiap bagian dari perangkat telah dapat bekerja sesuai dengan fungsi dan keinginan yang akan dibutuhkan.

2. Uji Unjuk Kerja

Pengujian unjuk kerja alat sangat dibutuhkan dengan tujuan agar dapat mengetahui apakah alat sesuai dengan standar pengukuran terhadap EKG yang ada. Beberapa hal yang perlu diamati antara lain :

H. Tabel Hasil Uji

A. Uji Fungsional

1. Pengujian tampilan modul EKG di PC

Tabel 3. Pengujian Tampilan Modul EKG di PC

No.	Display EKG di PC	Ket
1.		
2.		
3.		

2. Perbandingan EKG 3 Lead rumah sakit, dengan EKG 3 Lead

OLIMEX

Tabel 4. Perbandingan Pengujian EKG

No.	Jenis Display EKG	Display EKG
1.	Gelombang Standart EKG	
	EKG OLIMEX	
	Keterangan	
2.	Gelombang Standart EKG	
	EKG OLIMEX	

	Keterangan	
3.	Gelombang Standart EKG	
	EKG OLIMEX	
	Keterangan	

B. Uji Unjuk Kerja

1. Pengujian Unjuk Kerja Alat

Tabel 5. Pengujian Keakuratan Pembacaan BPM Alat

Percobaan	Pengukuran bpm dengan modul (bpm)	Keterangan
1		
2		
3		

2. Pengujian Unjuk Kerja Sistem

Tabel 6. Pengujian Unjuk Kerja Sistem

No.	Kondisi Power Supply dari laptop	Kondisi kabel USB	Kondisi Elektroda	Bentuk gelombang	keterangan
1	Stabil	Bagus	Baru		
2	Stabil	Bagus	Bekas		
3	Stabil	Jelek	Bekas		

I. Pengoperasian Alat

Pengoprasian alat ini dapat dilakukan dengan cara sebagai berikut:

1. Menancapkan sumber 5V ke *input* tegangan
2. Memasang eletroda pada titik yang telah ditentukan
3. Menjalankan software pada PC
4. Pilih com select, untuk memilih jalur komunikasi dengan arduino
5. Tekan tombol “continue”
6. Gelombang EKG akan tampil sesuai dengan setingan default.

BAB IV

PENGUJIAN DAN PEMBAHASAN

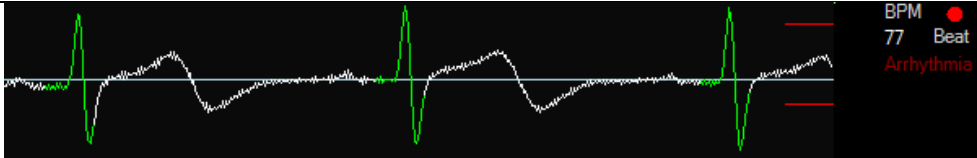
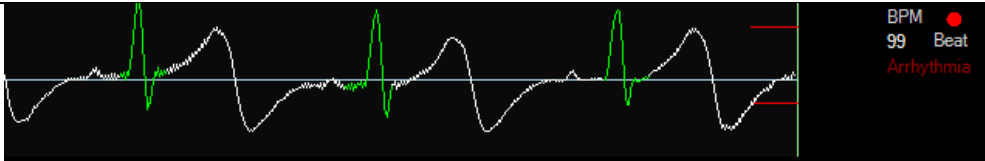
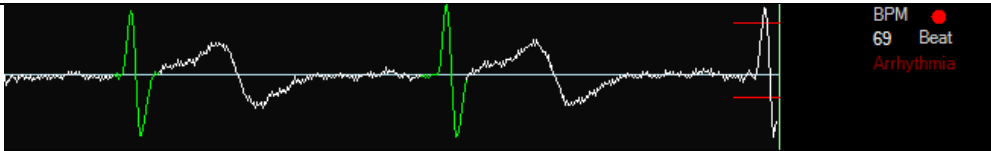
Tujuan dari pengujian dan pembahasan adalah untuk mengetahui kinerja alat yang sudah dibuat baik secara perbagian blok rangkaian maupun keseluruhan sistem, apakah sudah seperti yang diharapkan atau belum.

Pengujian ini meliputi:

A. Uji Fungsional

1. Pengujian tampilan modul EKG di PC

Tabel 7. Pengujian Tampilan Modul EKG di PC

No.	Display EKG di PC
1.	
	Ket : Gelombang muncul dan bpm terdeteksi
2.	
	Ket : Gelombang muncul dan bpm terdeteksi
3.	
	Ket : Gelombang muncul dan bpm terdeteksi

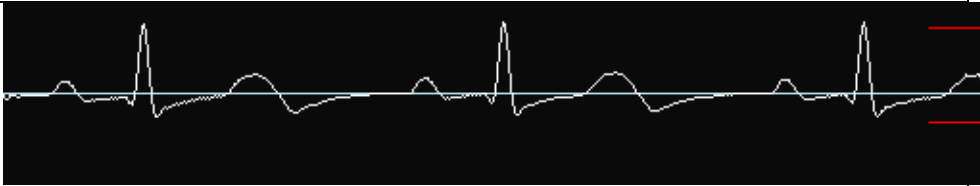
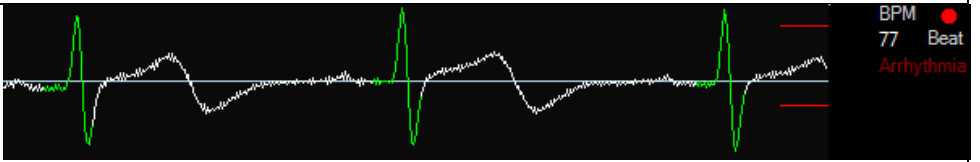
Tabel 7 ini berisikan tentang display/peraga gelombang EKG yang didapat dari hasil pemasangan electrode pada responden dengan kondisi normal. Hasil yang didapat direkam menggunakan software FreeHC.

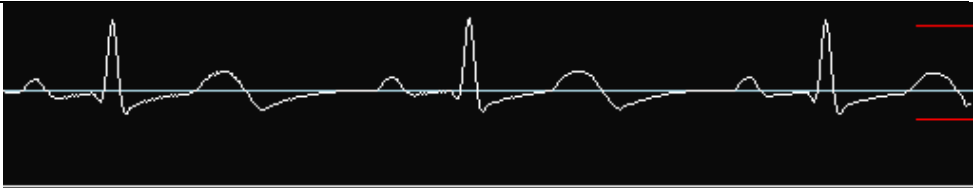
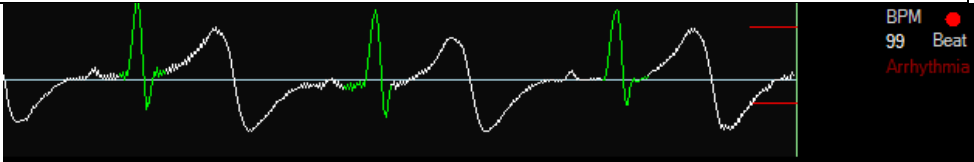
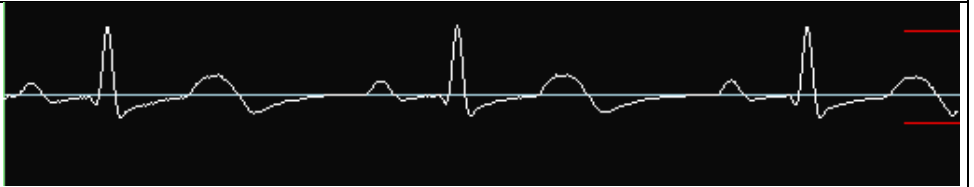
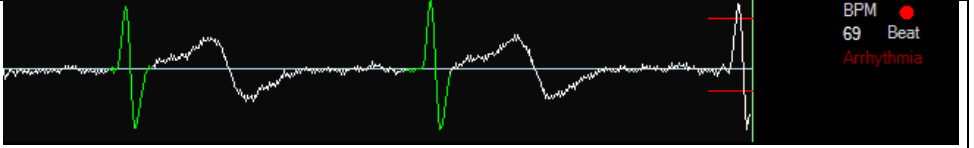
Pada percobaan diatas, hasil yang didapat dari perekaman sinyal jantung oleh EKG Shield OLIMEX menunjukkan hasil yang normal, gelombang jantung sudah dapat tampil dengan bagus, dan BPM dapat terdeteksi melalui software nya.

Percobaan pada tiap responded menghasilkan bentuk gelombang, dan BPM yang berbeda. Hal ini disebabkan oleh posisi pemasangan elektroda, posisi tubuh, dan kondisi responden pada saat melakukan pengambilan data. Berbagai posisi elektroda, dan tubuh responden saat melakukan pengambilan data, dapat dilihat dalam lampiran 2.

2. Perbandingan EKG 3 Lead rumah sakit, dengan EKG 3 Lead OLIMEX

Tabel 8. Perbandingan pengujian EKG

No.	Jenis Display EKG	Display EKG
1.	Gelombang Standart EKG	
	EKG OLIMEX	
	Keterangan	Gelombang Q, R ,T sesuai dan gelombang P S tdak sesuai

2.	Gelombang Standart EKG	
	EKG OLIMEX	
	Keterangan	Gelombang P, R, S mirip dan gelombang Q, T tidak sesuai
3.	Gelombang Standart EKG	
	EKG OLIMEX	
	Keterangan	Gelombang R dan T sesuai, P, Q, S tidak sesuai

Tabel 8 merupakan suatu perbandingan bentuk gelombang EKG standar dengan gelombang yang dihasilkan melalui electrode yang dipasangkan pada responden.

Gelombang EKG standar diperoleh dari pemasangan modul EKG dengan alat penguji standar EKG di rumah sakit untuk mengetahui apakah gelombang yang dihasilkan dari modul EKG sudah memenuhi syarat untuk dikatakan sebagai gelombang EKG. Syarat yang dimaksud adalah apakah bentuk gelombang EKG tersebut sesuai dengan ketentuan gelombang PQRST pada EKG sebenarnya, yang dimana jika diperhatikan pada tabel 8

dapat ditarik kesimpulan dimana 2 gelombang baik yang standar dan gelombang yang di dapat dari pemasangan elektroda pada responden akan terlihat sedikit berbeda pada bentuk gelombang PQRSST nya. Pada gelombang diatas, terlihat dari gambar hasil pemasangan elektroda pada responden masih terbilang kurang sempurna.

Terlihat pada kolom keterangan diatas, dalam 3 percobaan dengan membandingkan 2 gelombang EKG yang dihasilkan dari sinyal jantung yang berbeda, dengan menyamakan bpm keduanya, didapat bahwa ada beberapa gelombang PQRSST yang tidak sama diantara keduanya. Untuk percobaan pertama, membandingkan hasil gelombang EKG standar dengan BPM 80 dengan hasil gelombang EKG responden dengan BPM 77, hasil yang didapat yaitu kedua hasil gelombang mempunyai kesamaan dalam gelombang Q, R, dan T, sedangkan untuk gelombang P, dan S tidak memiliki kesamaan. Untuk percobaan kedua dan ketiga, hasil perbandingan yang didapat pun tidak sepenuhnya memiliki kesamaan, untuk percobaan kedua, perbandingan antara kedua hasil gelombang memiliki kesamaan pada gelombang P, R, dan S, sedangkan untuk gelombang Q, dan T tidak mempunyai kesamaan, dan untuk percobaan ketiga, hanya gelombang R, dan T yang memiliki kesamaan, sedangkan gelombang P, Q dan S tidak memiliki kesamaan sama. Berdasarkan hasil percobaan diatas dapat dibuat tabel perbandingan hasil gelombang yang memiliki kesamaan. Berikut tabel 9, tentang persamaan hasil gelombang antara 3 percobaan diatas :

No. Percobaan	Gelombang yang sama	Gelombang yang beda
1	Q, R, T	P, S
2	P, R, S	Q, T
3	R, T	P, Q, S

Tabel 9. Persamaan Hasil Gelombang

Dari tabel 9 dapat dibuat kesimpulan, gelombang R, dan T, dapat dikatakan sudah sesuai, dan gelombang P, Q, dan S, dapat dikatakan masih belum sesuai.

Ketidaksesuaian ini dapat disebabkan oleh penempatan elektroda pada responden yang tidak sesuai dengan titik sadap einhoven, dan juga kondisi tubuh responden ketika proses pengambilan data.

B. Uji Unjuk Kerja

1. Pengujian Unjuk Kerja Alat

Pengujian hasil kerja dilakukan untuk mengetahui kualitas dari kinerja seluruh rangkaian ini sebagai sebuah sistem secara utuh. Pengujian unjuk kerja dilakukan dari pengujian keakuratan pembacaan BPM alat. Pengujian ini bertujuan untuk mengetahui apakah alat sudah bekerja dengan baik atau masih terdapat kekurangan ketika melakukan proses kerja.

Tabel 10. Pengujian Keakuratan Pembacaan BPM Alat

Percobaan	Pengukuran bpm dengan modul (bpm)	Keterangan
1	77	Terdeteksi
2	99	Terdeteksi
3	69	Terdeteksi

Pada tabel 10 pengujian dilakukan dengan mengukur bpm, dengan menggunakan elektroda yang terpasang pada titik-titik sadapan pada badan manusia. Hasil yang diperoleh pada pengukuran tersebut tingkat bpm didapatkan dari hasil pembacaan sensor.

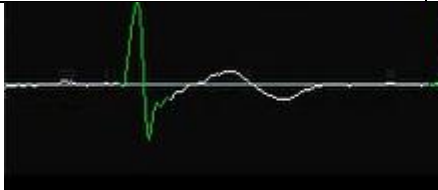
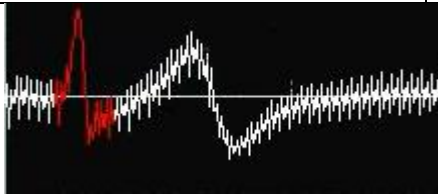
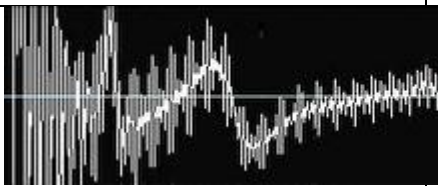
Pada pembacaan bpm dengan cara manual biasanya dilakukan dengan cara pengukuran detak jantung dari denyut nadi, dengan menghitung detak jantung selama 15 detik, setelah itu dikalikan dengan 4.

Untuk pengambilan yang efektif, dilakukan pengukuran bpm secara manual dan melalui modul secara bersamaan, untuk kemudian hasil yang tertera pada modul akan langsung diambil ketika pengukuran bpm secara manual selesai.

2. Pengujian Unjuk Kerja Sistem

Pengujian unjuk kerja sistem dilakukan untuk mengetahui apakah sistem alat bekerja sesuai dengan yang diharapkan atau tidak dengan menganalisis setiap bagiannya.

Tabel 11. Uji Alat Keseluruhan

No	Power Supply laptop	Kabel USB	Kondisi Elektroda	Bentuk gelombang	Keterangan
1.	Stabil	Bagus	Baru		Terbaca
2.	Stabil	Bagus	Bekas		Sinyal terdapat Noise
3.	Stabil	Jelek	Bekas		Tidak Terbaca

Tabel 11 merupakan hasil pengujian secara keseluruhan dengan berbagai faktor yang menentukan. Pengujian ini menentukan apakah gelombang yang tampil baik atau tidak

C. Pembahasan

1. Hardware

a. EKG SHIELD OLIMEX

EKG shield mempunyai *built-in* yang sudah disesuaikan dengan Arduino, sehingga rentang tegangan kerja alat menyerupai Arduino, dengan tegangan input 3,3V dan 5V, dengan sinyal kalibrasi juga disediakan dan dapat diakses lewat port D4/D9 digital output. Modul ini dapat dipasang hingga 6 buah shield secara paralel diatas nya, untuk mengaktifkan 6 channel pengamatan EKG.

b. Aduino MEGA 2560

Arduino MEGA 2560 merupakan modul mikrokontroler yang berbasis ATmega 2560, dan berfungsi sebagai pemroses data utama. Dengan tegangan input 7-12 V, dan tegangan output 6-20V, mempunyai 54 digital I/O pin, dan 16 analog pin. Inputan dari OLIMEX EKG akan disalurkan ke Arduino MEGA 2560 dengan baik.

2. Software

Dalam pembuatan program *software* yang digunakan adalah pemrograman Arduino UNO, menggunakan bahasa C/C++ berikut penjelasanya.

1. Switch_LED1

```
void Switch_LED1(void){  
  
    if((digitalRead(LED1))==HIGH){ digitalWrite(LED1,LOW); }  
  
    else{ digitalWrite(LED1,HIGH); }  
  
}
```

Fungsi *Switch_LED1(void)*, adalah untuk menhidupkan *Flashing* LED yang terletak pada arduino MEGA, digunakan sebagai sinyal penanda, jika kondisi LED1 == HIGH, dalam artian LED1 menyala, maka LED1 akan diset *LOW*, dalam artian mati, sedangkan *else{ digitalWrite(LED1,HIGH)*, akan mengembalikan kondisi LED1 pada keadaan *HIGH*, dalam artian menyala.

2. Switch_GAL_SIG

```
void toggle_GAL_SIG(void){  
    if(digitalRead(CAL_SIG) == HIGH){ digitalWrite(CAL_SIG,  
LOW); }  
    else{ digitalWrite(CAL_SIG, HIGH); }  
}
```

Program diatas memiliki arti kalibrasi sinyal, CAL_SIG sendiri merupakan variabel dari pada pin 9 dari OLIMEX shield tersebut. Fungsi diatas apabila dipanggil akan mengambil keputusan berdasarkan kondisi dari variabel CAL_SIG, yang mana jika variabel tersebut bernilai HIGH, maka variabel akan diubah nilainya menjadi LOW, akan tetapi jika kondisi variabel CAL_SIG tidak bernilai HIGH, maka variabel akan diubah nilai nya menjadi HIGH. Fungsi pin 9 sendiri sebagai kalibrasi sinyal merupakan fitur yang tersedia dari OLIMEX shield sendiri.

3. Setup

```
noInterrupts(); // Disable all interrupts before initialization  
  
// LED1  
pinMode(LED1, OUTPUT); //Setup LED1 direction
```

```

digitalWrite(LED1,LOW); //Setup LED1 state

pinMode(CAL_SIG, OUTPUT);

//Write packet header and footer
TXBuf[0] = 0xa5; //Sync 0
TXBuf[1] = 0x5a; //Sync 1
TXBuf[2] = 2; //Protocol version
TXBuf[3] = 0; //Packet counter
TXBuf[4] = 0x02; //CH1 High Byte
TXBuf[5] = 0x00; //CH1 Low Byte
TXBuf[6] = 0x02; //CH2 High Byte
TXBuf[7] = 0x00; //CH2 Low Byte
TXBuf[8] = 0x02; //CH3 High Byte
TXBuf[9] = 0x00; //CH3 Low Byte
TXBuf[10] = 0x02; //CH4 High Byte
TXBuf[11] = 0x00; //CH4 Low Byte
TXBuf[12] = 0x02; //CH5 High Byte
TXBuf[13] = 0x00; //CH5 Low Byte
TXBuf[14] = 0x02; //CH6 High Byte
TXBuf[15] = 0x00; //CH6 Low Byte

TXBuf[2 * NUMCHANNELS + HEADERLEN] = 0x01; //

Switches state

FlexiTimer2::set(TIMER2VAL, Timer2_Overflow_ISR);

FlexiTimer2::start();

```

```
Serial.begin(57600);
```

```
interrupts();
```

```
}
```

Program diatas berfungsi sebagai program utama dimana program tersebut menginisialisasikan seluruh variabel dan fungsi untuk eksekusi program. *noInterrupts* mempunyai fungsi untuk tidak memungkinkan beberapa fungsi untuk menjalankan beberapa program aktif pada *backgorund* dan beberapa komunikasi dapat diabaikan. Fungsi *pinMode* untuk menjadikan LED1 sebagai output dan dalam kondisi LOW maka LED1 mati lalu variabel CAL_SIG akan tersetting sebagai output.

Untuk selanjutnya bagian *TXBuf[]*; bagian ini berfungsi sebagai variabel array yang mana mendeklarasikan nilai-nilai dari tiap variabel array tersebut, agar satu nilai dapat ditempatkan dengan jelas. Sebagai contoh *TXBuf[4] = 0x02*;, yang berarti variabel array *TXBuf*, dengan array 4, mempunyai nilai 0x02. Dan begitu seterusnya. Secara keseluruhan blok deklarasi variabel array *TXBuf[]*; ini adalah untuk memudahkan dalam proses pembacaan, panulisan, dan pemrosesan data, yang mana dibutuhkan sebuah variabel array yang dapat dengan cepat dipanggil, dan ditulis ulang dengan input yang baru.

Kemudian ada fungsi yang dipanggil yaitu *FlexiTimer2::* yang berfungsi sebagai konfigurasi resolusi program, yang dalam program ini dapat diartikan sebagai, berapa detik sekali, input akan diupdate. Dalam program diatas *TIMER2VAL* merupakan variabel waktu yang mengatur sample

frekuensi yang digunakan untuk pengambilan data, dan *Timer2_Overflow_ISR* adalah sub-program yang akan dijalankan secara periodik berdasarkan waktu yang telah diatur di variabel *TIMER2VAL*. Setelah itu fungsi akan dijalankan dengan *FlexiTimer2::start()*;

Selanjutnya ialah penentuan kecepatan transfer data dari arduino ke PC yang di set dalam nilai 57600, yang berarti 57600 bit data per detik yang akan dikirimkan dari arduino ke PC. Pemilihan kecepatan transfer ini didasari oleh, $\text{ADC sampling rate} \times \text{packet size of OLIMEX} \times 10 = 43520$, yang mana ADC sampling rate sebesar 256, *packet size of OLIMEX* sebesar 17, dan 10 merupakan pembagian dari transmisi data yang terdiri dari 1 *start bit*, 8 *data bits*, dan 1 *stop bit*.

Yang terakhir adalah *interrupts()*; yang berfungsi untuk menghidupkan kembali interupt yang pada awalnya di tiadakan, agar data yang masuk dapat terbaca.

4. Timer2_Overflow_ISR

```
void Timer2_Overflow_ISR()
{

    Switch_LED1();

    for(CurrentCh=0;CurrentCh<6;CurrentCh++){

        ADC_Value = analogRead(CurrentCh);

        TXBuf[((2*CurrentCh) + HEADERLEN)] = ((unsigned
char)((ADC_Value & 0xFF00) >> 8));    // Write High Byte
```



```

        TXBuf[((2*CurrentCh) + HEADERLEN + 1)] = ((unsigned
char)(ADC_Value & 0x00FF));    // Write Low Byte

    }

    for(TXIndex=0;TXIndex<17;TXIndex++){

        Serial.write(TXBuf[TXIndex]);

    }

    TXBuf[3]++;

    counter++;

    if(counter == 12){

        counter = 0;

        toggle_GAL_SIG();

    }

}

```

Sub-program ini berfungsi sebagai pembacaan, dan pemrosesan ADC yang masuk. Pertama yaitu *Switch_LED1()*; , yang akan menghidupkan LED pada Arduino MEGA.

Untuk selanjutnya yaitu pembacaan input ADC, dengan menggunakan fungsi for. Pada bagian *for(CurrentCh=0;CurrentCh<6; CurrentCh++)* , CurrentCh akan bernilai 1 hingga 6, untuk selanjut nya dimasukkan dalam fungsi *ADC_Value = analogRead(CurrentCh);* , yang akan membaca nilai analog sesuai dengan nilai CurrnetCh, untuk kemudian dimasukkan dalam variabel ADC_Value. Pada baris selanjut nya ADC_Value akan di AND kan dengan

0xFF00 dan di *bitshift right*, yang akan menggeser nilai biner sebanyak 8 digit ke kanan, yang selanjutnya ini akan disebut dengan high byte, sedangkan untuk low byte, ADC_Value akan di AND kan dengan 0x00FF.

`for(TXIndex=0;TXIndex<17;TXIndex++)` adalah fungsi pengulangan untuk mengirimkan paket, dengan nilai TXIndex berawal dari 0 hingga 17, dengan penambahan 1 setiap pengulangan nya. Kemudian TXIndex akan dimasukkan dalam TXBuf dalam fungsi `Serial.write(TXBuf[TXIndex]);` yang mana nilai dari variabel TXbuf yang berada pada array TXIndex akan dikirimkan ke komunikasi serial untuk kemudian ditampilkan di PC.

Pada baris selanjutnya paket yang dikirim akan dihitung dengan fungsi `TXBuf[3]++;` yang mana nilai dari variabel array TXBuff[3] akan ditambahkan nilainya dengan 1.

`counter++;` akan menambahkan variabel counter dengan 1, dan kemudian akan dimasukkan dalam fungsi `if(counter == 12)` , yang akan menentukan jika variabel counter sudah berjumlah 12, maka `counter = 0;` , dan akan memanggil sub-program `toggle_GAL_SIG();`.

D. Cara Kerja Alat

Cara kerja dari alat ini pertama kali adalah dengan memasang elektroda pada titik sadapan yang telah ditentukan. Elektroda akan mengubah gelombang listrik menjadi sinyal gelombang jantung yang dihubungkan ke EKG shield Olimex dan akan ditampilkan pada komputer agar gelombang jantung yang di tampilkan dari hasil penyadapan Elektroda dapat dianalisa untuk mengetahui kelainan pada jantung pengguna.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan terhadap rancang bangun alat Elektrokardiograf (EKG) sebagai pemantau aktifitas kelistrikan jantung dengan menggunakan Arduino Mega 2560 dan EKG Shield, maka dapat disimpulkan:

1. Rancang bangun *hardware* pemantau gelombang kelistrikan aktifitas jantung telah terealisasi dengan menggunakan EKG Shield dan Arduino Mega 2560. Sistem pemantau menggunakan 3 buah Elektroda yang dipasangkan pada titik - titik tertentu sebagai pembaca dan pengkonversi sinyal gelombang jantung.
2. Rancang bangun *software* pemantau gelombang kelistrikan aktifitas jantung telah terealisasi dan berjalan dengan baik dengan menggunakan program berbasis bahasa C/++.
3. Unjuk kerja dari rancang bangun ini secara keseluruhan sudah berfungsi dengan baik. Dengan menempelkan Elektroda sesuai dengan titik yang ditentukan, gelombang yang ditampilkan pada monitor PC adalah gelombang Elektrokardiogram yang disertai dengan tampilan Bpm dari detak jantung.

B. Keterbatasan Alat

Rancang bangun alat Elektrokardiograf (EKG) sebagai pemantau aktifitas kelistrikan jantung dengan menggunakan Arduino Mega 2560 dan EKG Shield, memiliki keterbatasan dalam sistem kerjanya, antara lain:

1. Tidak ada sistem penyimpanan data manual ataupun otomatis
2. Tidak adanya *software* pengolah sinyal yang pasti untuk modul dari Olimex
3. Kurangnya datasheet mengenai modul EKG
4. Perlunya penambahan rangkaian Balance Differential Amplifire

C. Penelitian Lanjutan

Berdasarkan keterbatasan kemampuan dan waktu, penulis mengakui masih adanya kekurangan dalam pengerjaan alat yang dibuat ini, maka dari itu penulis menyarankan sebagai berikut:

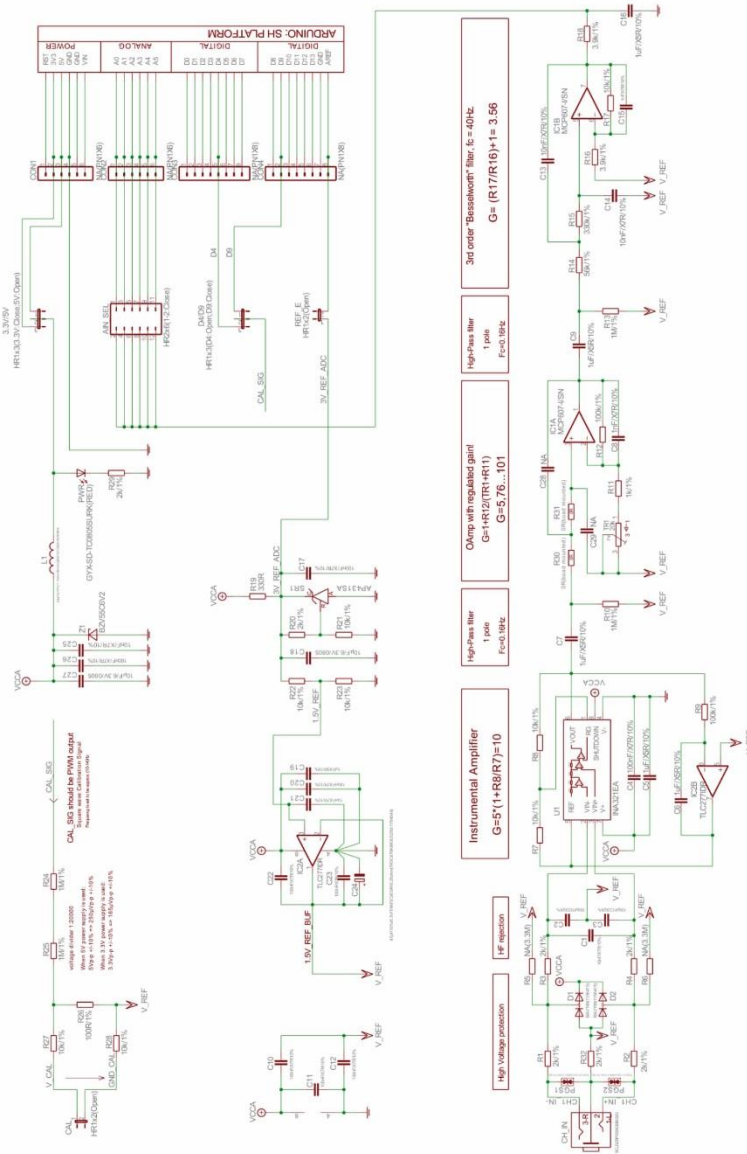
1. Penambahan modul untuk support penyimpanan data baik dengan menggunakan fitur telemetri berupa modul Ethernet Shield Arduino.
2. Penambahan modul EKG Shield untuk memperoleh hasil pembacaan yang lebih akurat.


DAFTAR PUSTAKA

- Mykclassroom. (2012). *Medical Electronics*. Diambil tanggal 23 Maret 2018 dari <https://www.mykclassroom.com/Engineering-branches/80/MEDICAL-ELECTRONICS>
- Rizal, A. (2014). *Instrumentasi Biomedis*. Yogyakarta: Graha Ilmu.
- Jptei. (2016). *Sejarah JPTEI UNY*. Diambil tanggal 8 Agustus 2018 dari <https://jptei.ft.uny.ac.id/sejarah>
- Kemdikbud. (2016). *Keputusan Direktur Jendral Pendidikan Dasar dan Menengah Kementerian Pendidikan dan Kebudayaan Nomor 4678, Tahun 2016, tentang Spektrum Keahlian Pendidikan Menengah Kejuruan*
- Prima P., Syandrez. (2011). *Membaca Hasil Ekg (Elektrokardiografi)*. Diambil tanggal 26 Maret 2018 dari <https://sandurezu.wordpress.com/2011/11/16/membaca-hasil-ekg-elektrokardiografi/>
- Farastya, P. (2017). *EKG, Mesin Pintar Pendeteksi Aktivitas Jantung*. Diambil tanggal 26 Maret 2018 dari <https://www.medicalogy.com/blog/ekg-mesin-pintar-pendeteksi-aktivitas-jantung/>
- Olimex. (2018). *Shield EKG EMG*. Diambil tanggal 25 Mei 2018 dari <https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/>
- Ndaru, D. (2017). *7 Fungsi EKG Jantung, Jenis, Prosedur, dan Biayanya*. Diambil tanggal 25 Mei 2018 dari <https://drjantung.com/fungsi-ekg-jantung>
- Johnag. (2017). *Arduino Portable EKG Monitor*. Diambil tanggal 5 Juni 2018 dari www.instructables.com/id/Arduino-Portable-EKG-Monitor/

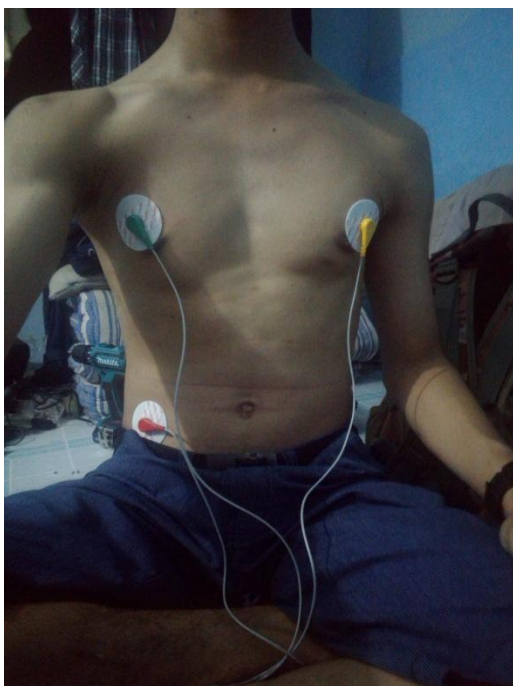
LAMPIRAN

A. Rangkaian EKG OLIMEX

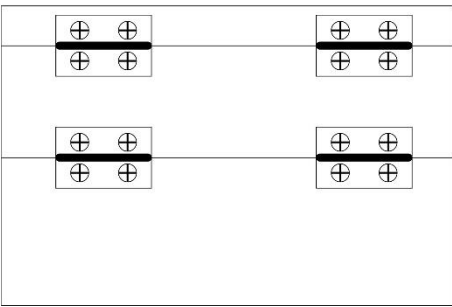
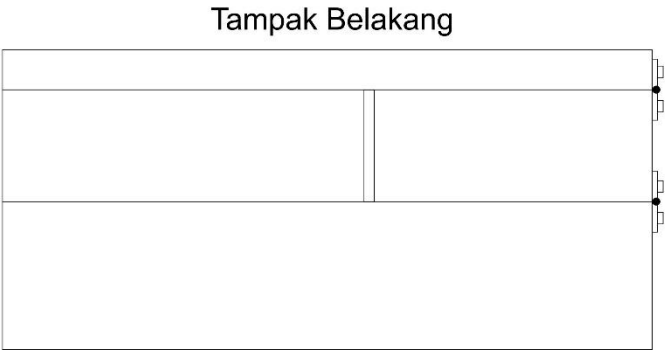
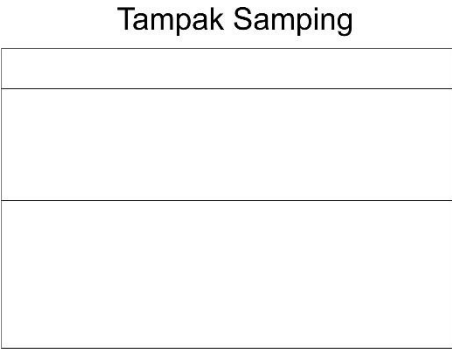
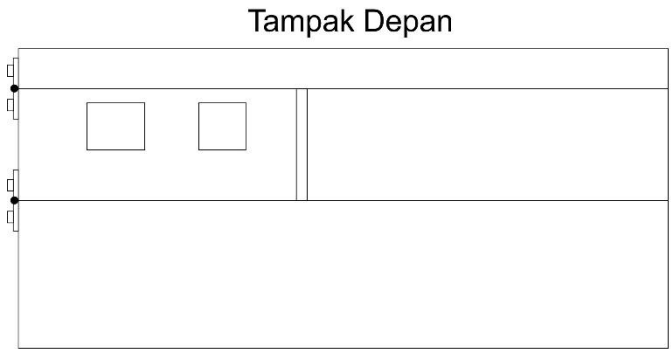




Lampiran 2. Foto pemasangan elektroda pada responden



Lampiran 3. Desain Box



Lampiran 4. Source code

```
#include <compat/deprecated.h>
#include <FlexiTimer2.h>

// All definitions
#define NUMCHANNELS 6
#define HEADERLEN 4
#define PACKETLEN (NUMCHANNELS * 2 + HEADERLEN + 1)
#define SAMPFREQ 256          // ADC sampling rate 256
#define TIMER2VAL (1024/(SAMPFREQ)) // Set 256Hz sampling frequency
#define LED1 13
#define CAL_SIG 9

// Global constants and variables
volatile unsigned char TXBuf[PACKETLEN]; //The transmission packet
volatile unsigned char TXIndex;          //Next byte to write in the transmission packet.
volatile unsigned char CurrentCh;        //Current channel being sampled.
volatile unsigned char counter = 0;      //Additional divider used to generate CAL_SIG
volatile unsigned int ADC_Value = 0;      //ADC current value

//~~~~~
// Functions
//~~~~~

/*****
/* Function name: Toggle_LED1          */
/* Parameters          */
/* Input : No          */
/* Output : No         */
/* Action: Switches-over LED1.        */
*****/

void Toggle_LED1(void){
```

```

if((digitalRead(LED1))==HIGH){ digitalWrite(LED1,LOW); }
else{ digitalWrite(LED1,HIGH); }

}

/*****
/* Function name: toggle_GAL_SIG          */
/* Parameters                          */
/* Input : No                          */
/* Output : No                        */
/* Action: Switches-over GAL_SIG.      */
*****/
void toggle_GAL_SIG(void){

if(digitalRead(CAL_SIG) == HIGH){ digitalWrite(CAL_SIG, LOW); }
else{ digitalWrite(CAL_SIG, HIGH); }

}

/*****
/* Function name: setup                  */
/* Parameters                          */
/* Input : No                          */
/* Output : No                        */
/* Action: Initializes all peripherals */
*****/
void setup() {

noInterrupts(); // Disable all interrupts before initialization

// LED1
pinMode(LED1, OUTPUT); //Setup LED1 direction
digitalWrite(LED1,LOW); //Setup LED1 state

```

```

pinMode(CAL_SIG, OUTPUT);

//Write packet header and footer
TXBuf[0] = 0xa5; //Sync 0
TXBuf[1] = 0x5a; //Sync 1
TXBuf[2] = 2; //Protocol version
TXBuf[3] = 0; //Packet counter
TXBuf[4] = 0x02; //CH1 High Byte
TXBuf[5] = 0x00; //CH1 Low Byte
TXBuf[6] = 0x02; //CH2 High Byte
TXBuf[7] = 0x00; //CH2 Low Byte
TXBuf[8] = 0x02; //CH3 High Byte
TXBuf[9] = 0x00; //CH3 Low Byte
TXBuf[10] = 0x02; //CH4 High Byte
TXBuf[11] = 0x00; //CH4 Low Byte
TXBuf[12] = 0x02; //CH5 High Byte
TXBuf[13] = 0x00; //CH5 Low Byte
TXBuf[14] = 0x02; //CH6 High Byte
TXBuf[15] = 0x00; //CH6 Low Byte
TXBuf[2 * NUMCHANNELS + HEADERLEN] = 0x01; // Switches state

// Timer2
// Timer2 is used to setup the analog channels sampling frequency and packet update.
// Whenever interrupt occurs, the current read packet is sent to the PC
// In addition the CAL_SIG is generated as well, so Timer1 is not required in this case!
FlexiTimer2::set(TIMER2VAL, Timer2_Overflow_ISR);
FlexiTimer2::start();

// Serial Port
Serial.begin(57600);
//Set speed to 57600 bps

// MCU sleep mode = idle.
//outb(MCUCR,(inp(MCUCR) | (1<<SE)) & ~(1<<SM0) | ~(1<<SM1) | ~(1<<SM2)));

```

```

interrupts(); // Enable all interrupts after initialization has been completed
}

/*****
/* Function name: Timer2_Overflow_ISR      */
/* Parameters          */
/* Input   : No          */
/* Output  : No          */
/* Action: Determines ADC sampling frequency. */
*****/

void Timer2_Overflow_ISR()
{
    // Toggle LED1 with ADC sampling frequency /2
    Toggle_LED1();

    //Read the 6 ADC inputs and store current values in Packet
    for(CurrentCh=0;CurrentCh<6;CurrentCh++){
        ADC_Value = analogRead(CurrentCh);
        TXBuf[((2*CurrentCh) + HEADERLEN)] = ((unsigned char)((ADC_Value & 0xFF00) >> 8)); //
        Write High Byte
        TXBuf[((2*CurrentCh) + HEADERLEN + 1)] = ((unsigned char)(ADC_Value & 0x00FF)); //
        Write Low Byte
    }

    // Send Packet
    for(TXIndex=0;TXIndex<17;TXIndex++){
        Serial.write(TXBuf[TXIndex]);
    }

    // Increment the packet counter
    TXBuf[3]++;

    // Generate the CAL_SIGnal
    counter++; // increment the divider counter
    if(counter == 12){ // 250/12/2 = 10.4Hz ->Toggle frequency

```

```

counter = 0;
toggle_GAL_SIG();    // Generate CAL signal with frequ ~10Hz
}
}

/*****
/* Function name: loop          */
/* Parameters                */
/* Input  : No                */
/* Output : No                */
/* Action: Puts MCU into sleep mode. */
*****/
void loop() {

__asm__ __volatile__ ("sleep");

}

```

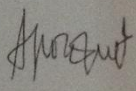
Lampiran 5. Scan Form Kuesioner

Nama : AGUS SUSILO WIBOWO, ST

Pendapat anda tentang alat ini :

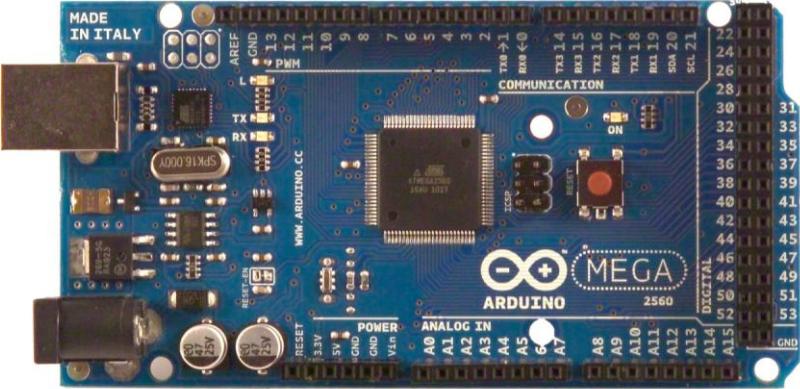
Untuk simulasi alat sudah terlampir normal.

Tanda tangan



Lampiran 6. Datasheet Arduino Mega 2560

Arduino MEGA 2560



MADE IN ITALY






ARDUINO MEGA 2560

CE

Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

	Index
Technical Specifications	Page 2
How to use Arduino Programming Enviroment, Basic Tutorials	Page 6
Terms & Conditions	Page 7
Enviromental Policies half sqm of green via Impatto Zero®	Page 7



Technical Specification

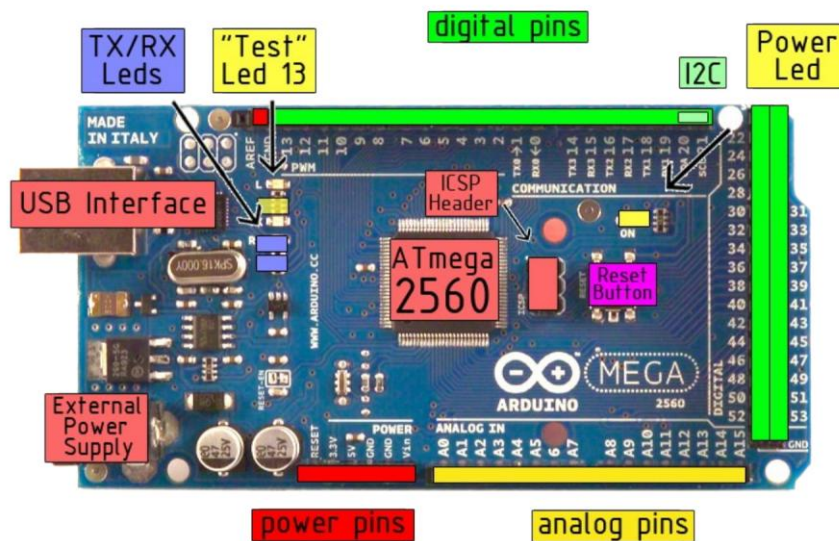


EAGLE files: [_arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

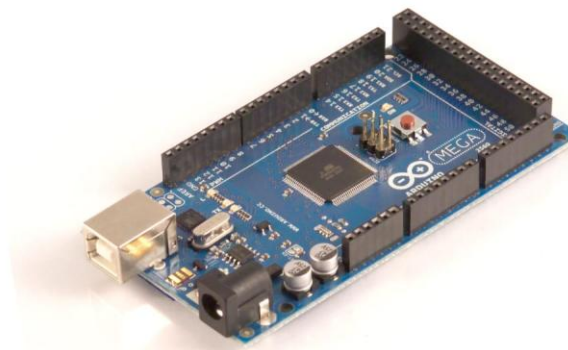
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop() {
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // set the LED off
  delay(1000);                // wait for a second
}
```



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!

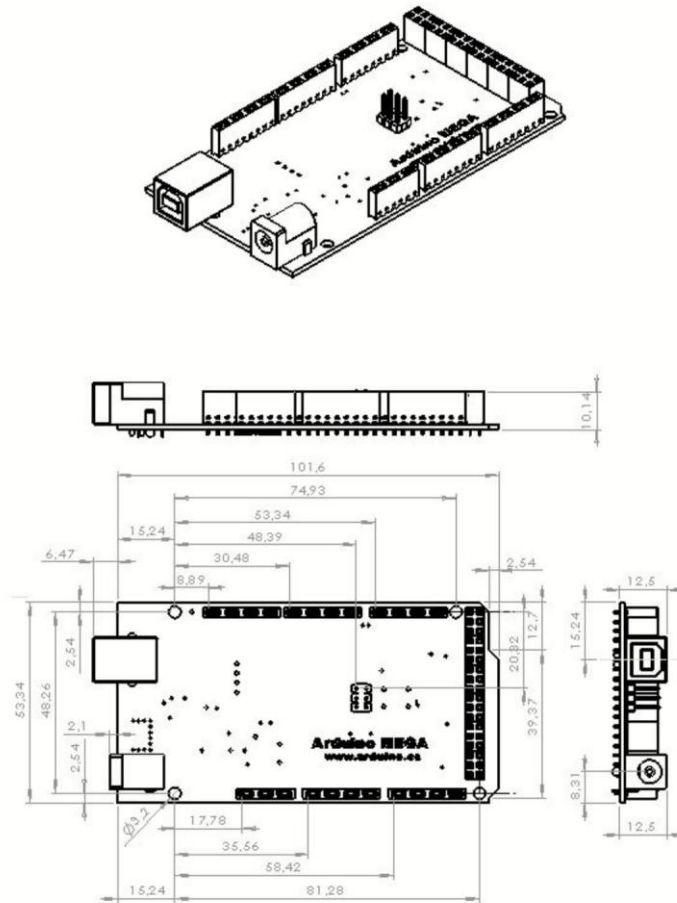


radiospares

RADIONICS



Dimensioned Drawing



radiospares RADIONICS



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Enviromental Policies



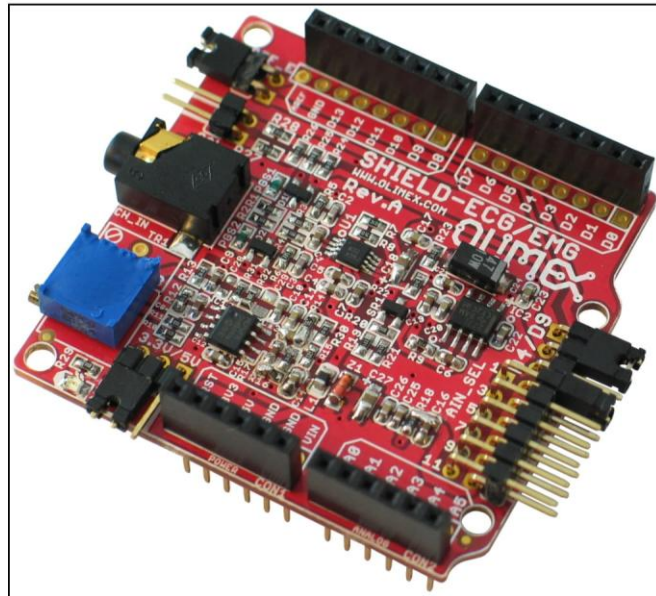
The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS





SHIELD-EKG-EMG bio-feedback shield **USER'S MANUAL**

All boards produced by Olimex LTD are ROHS compliant



Revision E, June 2014
Designed by OLIMEX Ltd, 2011

DISCLAIMER

© 2013 Olimex Ltd. Olimex®, logo and combinations thereof, are registered trademarks of Olimex Ltd. Other product names may be trademarks of others and the rights belong to their respective owners.

The information in this document is provided in connection with Olimex products. No license, express or implied or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Olimex products.

This work is licensed under the Creative Commons Attribution-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://www.creativecommons.org/licenses/by-sa/3.0/>.



This hardware design by Olimex LTD is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License.

The software is released under GPL.

It is possible that the pictures in this manual differ from the latest revision of the board.

The product described in this document is subject to continuous development and improvements. All particulars of the product and its use contained in this document are given by OLIMEX in good faith. However all warranties implied or expressed including but not limited to implied warranties of merchantability or fitness for purpose are excluded. This document is intended only to assist the reader in the use of the product. OLIMEX Ltd. shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

This evaluation board/kit is intended for use for engineering development, demonstration, or evaluation purposes only and is not considered by OLIMEX to be a finished end-product fit for general consumer use. Persons handling the product must have electronics training and observe good engineering practice standards. As such, the goods being provided are not intended to be complete in terms of required design-, marketing-, and/or manufacturing-related protective considerations, including product safety and environmental measures typically found in end products that incorporate such semiconductor components or circuit boards.

Olimex currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. Olimex assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

THERE IS NO WARRANTY FOR THE DESIGN MATERIALS AND THE COMPONENTS USED TO CREATE SHIELD-EKG-EMG. THEY ARE CONSIDERED SUITABLE ONLY FOR SHIELD-EKG-EMG.

WARNING:**EKG DEVICE DISCLAIMER**

IEC601 is a standard that specifies tests and requirements that medical devices must pass before they can be used on humans. However, none of the devices built from these designs have been tested according to these guidelines because of the costs involved. Therefore, a device based on any of these designs may not be used for medical purposes as no medical claims are made.

NOTE THAT CONNECTING A DEVICE VIA ELECTRODES TO HUMANS OR ANIMALS IS POTENTIALLY HAZARDOUS AND MAY RESULT IN ELECTRIC SHOCK AND/OR SEIZURE.

Table of Contents

DISCLAIMER.....	2
SECTION 1 OVERVIEW.....	5
1.1 Scope.....	5
1.2 Features.....	5
1.3 Similar boards.....	5
1.4 Organization.....	6
SECTION 2 SETTING UP THE SHIELD-EKG-EMG.....	7
2.1 Electrostatic Warning.....	7
2.2 Requirements.....	7
2.3 Powering up the board.....	8
2.4 Arduino/Maple/Pinguino note.....	8
SECTION 3 SHIELD-EKG-EMG BOARD DESCRIPTION.....	10
3.1 Layout (Top view):.....	10
3.2 Layout (Bottom view):.....	11
SECTION 4 INSTALLATION EXAMPLE.....	12
4.1 SHIELD-EKG-EMG and OLIMEXINO-328.....	12
SECTION 5 HARDWARE.....	14
5.1 Arduino shield connector.....	14
5.2 Trimmer TR1.....	14
5.3 Jumper description.....	14
5.4 Custom electrode connection.....	16
SECTION 6 SCHEMATICS.....	17
6.1 Eagle schematics.....	17
6.2 Physical dimensions.....	19
SECTION 7 REVISION HISTORY.....	20
7.1 Document revision.....	20
7.2 Web page of your device.....	20
7.3 Ordering info.....	20

SECTION 1 OVERVIEW

Thank you for choosing the SHIELD-EKG-EMG prototype board from Olimex! This document provides a User's Guide for the Olimex SHIELD-EKG-EMG. As an overview, this chapter gives the scope of this document and lists the board's features. The document's organization is then detailed.

1.1 Scope

The SHIELD-EKG-EMG is an extension module for Olimex's ARDUINO compatible boards – like OLIMEXINO-328, OLIMEXINO-STM32 and PIC32-PINGUINO, among others. The shield is also compatible with ARDUINO boards including ARDUINO UNO. The board comes with mounted connectors on it.

1.2 Features

- Suitable for electrocardiography – monitoring and data collection
- Suitable for electromyography – monitoring and data collection
- Stackable shield (pass-through headers) - up to 6 shields forming 6 channels; mounted on top of each other and wired to A0-A6 analogue inputs
- Open hardware, open software project – users have access to all design documents
- Calibration signal generation by D4/D9 digital output
- Precise trimmer potentiometer for calibration (all boards are shipped completely assembled, tested and calibrated so you should not do this unless you want to see how things work)
- Input connector for passive or active electrodes
- Works with both 3.3V and 5V Arduino boards
- Available examples for Arduino, Pinguino and Maple

Please note that the unit can be used for electroencephalography to some extent, however such use is not intended by design. The design doesn't provide the accuracy required for proper EEG and if you are interested in this field, please, consider the EEG-SMT board.

1.3 Similar boards

EEG-SMT – an inexpensive unit specifically designed for electroencephalography (EEG) - recording of electrical activity along the scalp (brain activity):
<https://www.olimex.com/Products/EEG/OpenEEG/EEG-SMT/>

PROTO-SHIELD – If you seek a board which follows the DUINO shield dimensions specification and offers prototype area and 2 buttons with filter capacitors and 2 LEDs. Web page of the product: <https://www.olimex.com/Products/Duino/Shields/PROTO-SHIELD/>

SHIELD-LOL – If you are searching for a shield with 126 LED matrix (14×9) available in 4 colors and 4 different sizes of the LEDs. Web page:
<https://www.olimex.com/Products/Duino/Shields/SHIELD-LOL/>

1.4 Organization

Each section in this document covers a separate topic, organized as follow:

- Section 1 is an overview of the board usage and features
- Section 2 provides a guide for quickly setting up the board and introduces the user to Arduino/Maple/Pinguino
- Section 3 contains the general board diagram and layout
- Section 4 show and example of setting up SHIELD-EKG-EMG with OLIMEXINO-328
- Section 5 covers the connector pinout, peripherals and jumper description
- Section 6 provides the schematics
- Section 7 contains the revision history

SECTION 2 SETTING UP THE SHIELD-EKG-EMG

This section helps you set up the SHIELD-EKG-EMG development board for the first time.

Please consider first the electrostatic warning to avoid damaging the board, then discover the hardware and software required to operate the board.

The procedure to power up the board is given, and a description of the default board behavior is detailed.

2.1 Electrostatic Warning

The SHIELD-EKG-EMG development board is shipped in a protective anti-static package. The board must not be exposed to high electrostatic potentials. A grounding strap or similar protective device should be worn when handling the board. Avoid touching the component pins or any other metallic element.

2.2 Requirements

In order to set up the SHIELD-EKG-EMG prototype board, the following items are required:

- SHIELD-EKG-EMG itself
- ARDUINO compatible board* (e.g. OLIMEXINO-328, OLIMEXINO-STM32, PIC32-PINGUINO)
- Electrode cable**

*The pinout for the board strictly follows the DUINO extension specification. Best choice for a board would be OLIMEXINO-328 or any DUINO board which utilizes the ATmega328 since we have working and tested examples for those MCUs, configuring them for ATmega128 would require tweaking of the code.

**The cable features three electrodes – two data electrodes (1 channel) and DLR electrode (feedback). If you use more than one SHIELD-EKG-EMG you can use cables without DLR for every shield after the first.

IMPORTANT NOTE: The electrode cables for SHIELD-EKG-EMG and MOD-EEG-SMT are different and not compatible!

There are two electrode cables that we have tested with the board. They are named SHIELD-EKG-EMG-PA and SHIELD-EKG-EMG-PRO. The first one is considered open-hardware and its schematics might be used as a reference if you wish to make the cable yourself.

The SHIELD-EKG-EMG-PRO works with different set of attachment cups that makes it easier to measure EMG signals at hard-to-reach spots and also allow contact materials replacement.

Links to the web-pages of the electrodes:

- <https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG-PA/open-source->

hardware

- <https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG-PRO/>

Different pads for SHIELD-EKG-EMG-PRO electrode:

- <https://www.olimex.com/Products/Modules/Biofeedback/ECG-GEL-ELECTRODE/>

- <https://www.olimex.com/Products/Duino/Shields/ECG-CLIP/>

- <https://www.olimex.com/Products/Duino/Shields/ECG-SCUP/>

In case you want to build a working SHIELD-EKG-EMG system without owning a DUNIO board the best choice would be our board OLIMEXINO-328. This is the web page for the board: <https://www.olimex.com/Products/Duino/AVR/OLIMEXINO-328/>.

2.3 Powering up the board

The SHIELD-EKG-EMG board is powered by the host board it is mounted on. There is the option to be powered either by 3.3V or 5.0V host board (configured easily by a jumper).

On powering the board PWR LED must become RED.

2.4 Arduino/Maple/Pinguino note

What is Arduino?

Arduino is an open-source electronics prototyping platform, designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open hardware design for the Arduino board with an Atmel AVR processor and on-board I/O support. The software consists of a standard programming language and the boot loader that runs on the board.

Arduino hardware is programmed using a Wiring-based language (syntax + libraries), similar to C++ with some simplifications and modifications, and a Processing-based Integrated Development Environment (IDE).

The project began in Ivrea, Italy in 2005 aiming to make a device for controlling student-built interaction design projects less expensively than other prototyping systems available at the time. As of February 2010 more than 120,000 Arduino boards had been shipped. Founders Massimo Banzi and David Cuartielles named the project after a local bar named Arduino. The name is an Italian masculine first name, meaning "strong friend". The English pronunciation is "Hardwin", a namesake of Arduino of Ivrea.

More information could be found at the creators web page <http://arduino.cc/> and in the Arduino Wiki <http://en.wikipedia.org/wiki/Arduino>.

To make the story short – Arduino is easy for beginners who lack Electronics knowledge, but also does not restrict professionals as they can program it in C++ or mix of Arduino/C++ language.

There are thousands of projects which makes it easy to startup as there is barely no field

where Arduino enthusiasts to have not been already.

Arduino has inspired two other major derivatives – MAPLE and PINGUINO. Based on 8-bit AVR technology the computational power of Arduino boards is modest, this is why a team from MIT developed the MAPLE project which is based on ARM7 STM32F103RBT6 microcontroller. The board have same friendly IDE as Arduino and offers the same capabilities as hardware and software but runs the Arduino code much faster. The Maple project can be found at <http://leaflabs.com>

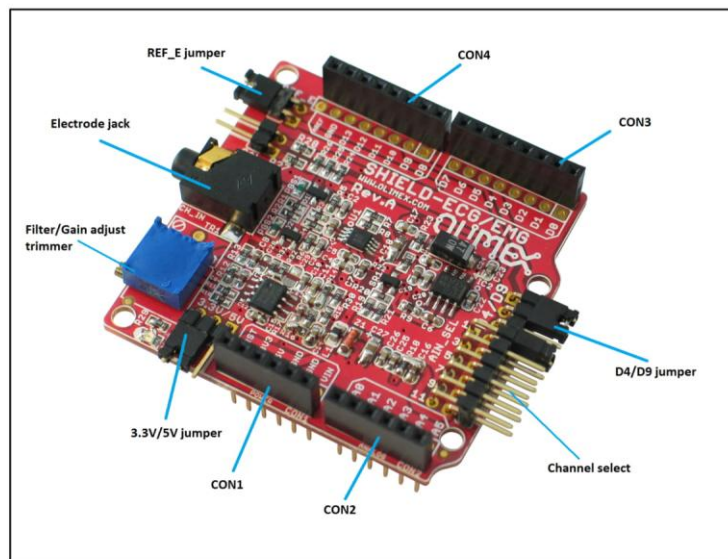
In parallel with Arduino another project was started called PINGUINO. This project chose its first implementation to be with PIC microcontrollers, as AVR's were hard to find in some parts of the world like South America so it is likely to see lot of PINGUINO developers are from that part of the world. PINGUINO project founders decided to go with Python instead Java for processing language. For the moment PINGUINO is much more flexible than Arduino as it is not limited to 8bit microcontrollers. Currently the IDE, which has GCC in background, can support 8-bit PIC microcontrollers, 32bit PIC32 (MIPS) microcontrollers and ARM7/CORTEXM3 microcontrollers which makes PINGUINO very flexible because once you make your project you can migrate easily through different hardware platforms and not being bound to a single microcontroller manufacturer. The PINGUINO project can be found at: <http://www.pinguino.cc>.

SECTION 3 SHIELD-EKG-EMG BOARD DESCRIPTION

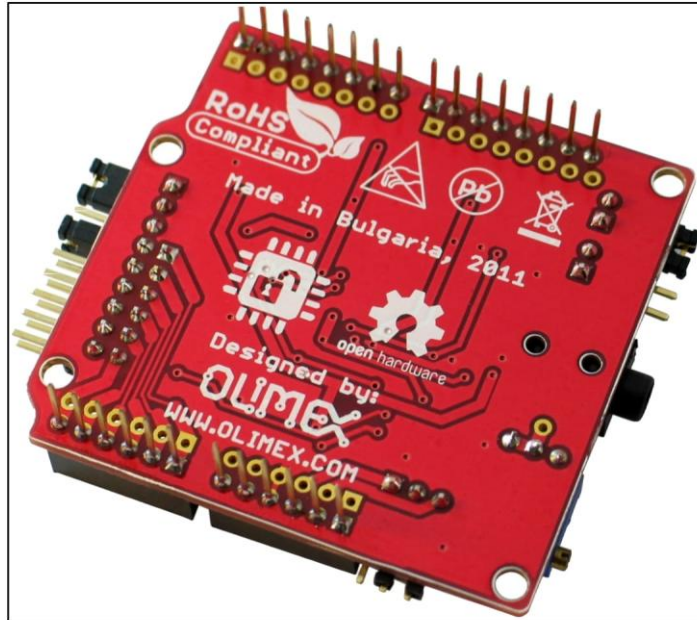
Here you get acquainted with the main parts of the board. Note the names used on the board differ from the names used to describe them. For the actual names check the SHIELD-EKG-EMG board itself.

For example: BUTTON (seen on the op view below) is named BUT; RESET is named RST; etc

3.1 Layout (Top view):



3.2 Layout (Bottom view):



SECTION 4 INSTALLATION EXAMPLE

This is a step by step example of installing SHIELD-EKG-EMG on OLIMEXINO-328 using Windows. You can refer to the tips keeping in mind that the example utilizes a board with ATmega328 MCU.

4.1 SHIELD-EKG-EMG and OLIMEXINO-328

In this example we use OLIMEXINO-328; SHIELD-EKG-EMG; USB – USB mini cable; Arduino 1.0 IDE; two external libraries for the IDE (the latest versions of TimerOne9 and FlexiTimer2); the latest FTDI VCP drivers (2.08.14), demo code provided by us that can be downloaded from the web site and free monitoring software Electric Guru.

0. IMPORTANT! Before starting you have to prepare the OLIMEXINO-328 board by removing its wire between the R6 pads (R6 is not mounted). There is increased power consumption and a chance of electrical failure if you omit to do so. This is only for the OLIMEXINO-328 board! Don't do the modification if you lack basic electronics skills, because it is possible to damage the board.

If the host board provides reference voltage on the AREF pin then open REF_E jumper. We don't want two supplies to provide power to the same line – the 3V_REF_ADC one.

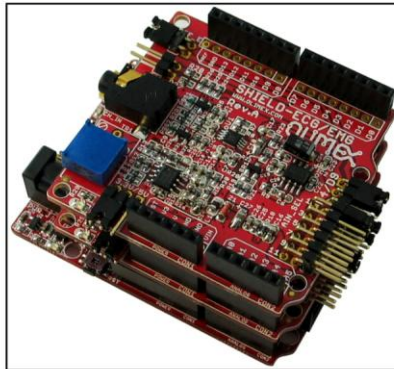
1. Download and extract the Arduino 1.1 IDE package from the Arduino web site:
<http://www.arduino.cc/>
2. Download and place the two timer libraries required (TimerOne, FlexiTimer2) in \arduino-1.0\libraries by placing each of them in properly named folder (check the other libraries for reference)
<http://arduino.cc/playground/Code/Timer1>
http://www.pjrc.com/teensy/td_libs_MsTimer2.html
3. Download the demo project from product's web page:
<https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/>
4. Download the FTDI VCP drivers: <http://www.ftdichip.com/Drivers/VCP.htm>

5. Set the jumpers of SHIELD-EKG-EMG in the following way:

REF_E – closed
3.3V/5V – 5V position
D4/D9 – D9 position
ANI_SEL – 1 position (channel)

6. Connect the shield to the board

7. Connect OLIMEXINO-328 to the USB



8. Install the VCP FTDI drivers by going in *Device Manager*; right-clicking over the

unrecognized device and choosing *Update Driver* and then pointing to the folder where you downloaded and extracted the FTDI VCP driver.
Here it is advisable to go to *Device Manager* and from advanced settings of our recognized USB Serial Port (COMx) device to set x to a free port between 1 and 4 (because the monitoring software in this example can read only from COM ports 1 to 4).

9. Start Arduino IDE and open the provided by Olimex project
ShieldEkgEmgDemo.pde

10. Set Tools -> Board -> Arduino Duemilanove w/ ATmega328
Set Tools -> Serial port -> the COM we configured our board at

11. Click Upload (→)

12. Download, install and start the software from this page:

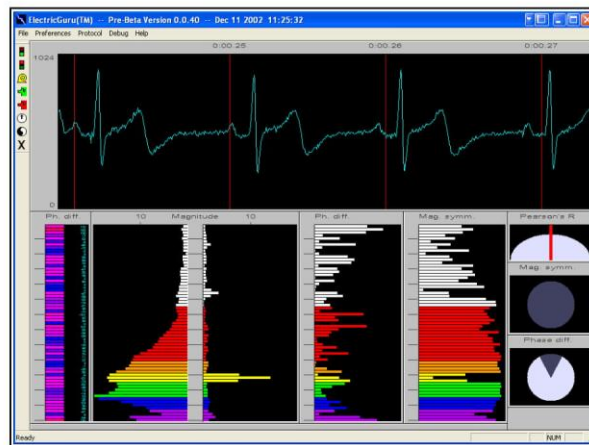
<https://www.olimex.com/Products/EEG/OpenEEG/EEG-SMT/resources/ElecGuru40.zip>

13. It is advisable to adjust the settings in your ElecGuru program in Preferences-> Trace (waveform)... (depending how many channels/shields you use)

14. Choose the COM port your OLIMEXINO is connected to from Preferences -> Serial port... (Remember you have to set it to COM 1 to 4)

15. You connect the electrodes to your right arm, left arm and the DLR to your right leg

16. Start monitoring, adjust the settings until you receive an image like the one shown on the picture



SECTION 5 HARDWARE

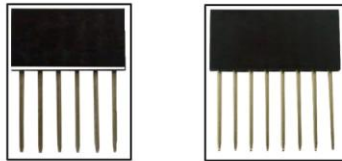
You can get a good view of the hardware observing the board. All pins, connectors and jumpers are named individually.

5.1 Arduino shield connector

These connectors follow the ARDUINO specification for shield connection. The shield comes with soldered connectors making it ready for mounting on compatible board with the possibility to have another shield mounted on it.

Pin #	POWER CON1	ANALOG CON2	DIGITAL CON3	DIGITAL CON4
1	RST	A0	D0	D8
2	3.3V	A1	D1	D9
3	5V	A2	D2	D10
4	GND	A3	D3	D11
5	GND	A4	D4	D12
6	Vin	A5	D5	D13
7	-	-	D6	GND
8	-	-	D7	AREF

6-pin and 8-pin connectors mounted (CON1, CON2 and CON3, CON4):



5.2 Trimmer TR1

Trimmer TR1 is calibrated during the factory testing. However it may be adjusted for the gain. Use at own risk.

5.3 Jumper description

The names of the jumpers on the board correspond to the bold names used below:

3.3V/5V

This jumper controls the power circuit. Whether powered by 3.3V or 5V board.

Default state is 3.3V.

REF_E

The position of the REF_E jumper depends on the "host" board. If the "host" board provides voltage on the AREF pin of the digital connector REF_E has to be open. If there is no voltage provided on the AREF pin then SHIELD-EKG-EMG's REF_E jumper has to be closed.

If both boards provide reference voltage to the 3V_REF_ADC line (and AREF pin respectively) then there would be a circuit conflict which might damage a component on one of the boards.

You need to ensure only one of the boards provides AREF voltage.

If you use only one shield this jumper has to be closed.

If you have multiple shields the first one should be closed; the respective REF_E jumpers on every other shield above it should be open.

Default state is closed.

AIN_SEL

This jumper is responsible for which channel the current SHIELD-EKG-EMG would utilize. If you have more than one shield one of them should have AIN_SEL in position 1, the second in position 3, etc.

Default state is in position 1.

D4/D9

Controls pin D4/D9. Some processors utilize the default D9 pin so you have to switch to D4. This jumper provides easy option to do so.

Default state is D9.

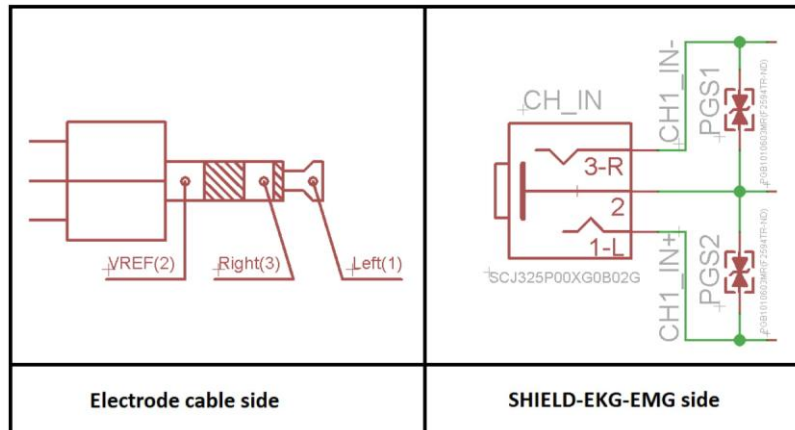
CAL

CAL jumper is used for feedback of the calibration and requires additional cable.

Default state is open.

5.4 Custom electrode connection

To make the passive electrode yourself you need three cables, passive electrode surface and audio jack. The way the signals go for connecting the audio jack is shown below:



Note that connecting active electrode that way will probably lead to electrical shock for SHIELD-EKG-EMG. Only passive electrodes can be used with SHIELD-EKG-EMG.

SECTION 6 SCHEMATICS

6.1 Eagle schematics

SHIELD-EKG-EMG schematic is visible for reference here. But you can find it with better resolution and stand-alone on the product's web page:

<https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/>. They are located in HARDWARE section.

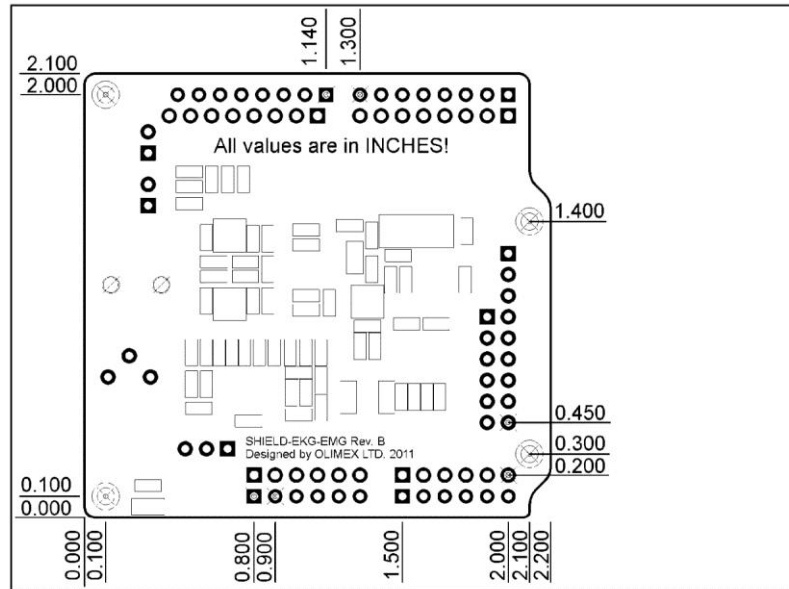
The EAGLE schematic is situated on the next page for quicker reference.



6.2 Physical dimensions

As you can see below the dimensions follow the classic Arduino shield pin specification making it compatible with all Olimex Duino boards.

Note: All dimensions are in inches!



SECTION 7 REVISION HISTORY

7.1 Document revision

Revision	Changes	Modified Pages
A	Initial Creation	All
B	Added important note on removing a wire at R6 pads from OLIMEXINO-328	11.12
C	Fixed error on page 7 about the LED color Added note on page 12 about possible damage due to lack of experience Changed schematic with a searchable one Various formatting adjustments	7, 12, 16
D	Changed disclaimer section to represent the open-source nature of the shield Added explanation of the cable	2, 15
E	Changed formatting of the document Updated links Added few new notes on REF_E jumper	All

7.2 Web page of your device

The web page you can visit for more info on your device is
<https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/>.

There are examples and libraries available for Arduino IDE, Pinguino IDE and Maple IDE.

7.3 Ordering info

ORDER CODES:

SHIELD-EKG-EMG – completely assembled and tested

OLIMEXINO-328 – Arduino based board with ATmega328, fully compatible with SHIELD-EKG-EMG

How to order?

You can order directly from our online shop or from any of our distributors. The list of distributors might be found here: <https://www.olimex.com/Distributors>.

Check our web site <https://www.olimex.com/> for more info.